From predictive to prescriptive process monitoring: Recommending the next best actions instead of calculating the next most likely events

Sven Weinzierl[1], Sandra Zilker[1], Matthias Stierle[1], Gyunam Park[2], and Martin Matzner[1]

[1] Friedrich-Alexander Universität Erlangen-Nürnberg, Digital Industrial Service Systems, Nuremberg, Germany; [2] Pohang University of Science and Technology, Analytics & Information Management Lab, Pohang, South Korea {sven.weinzierl; sandra.zilker; matthias.stierle; martin.matzner}@fau.de, {gnpark}@postech.ac.kr

Abstract. Predictive business process monitoring (PBPM) deals with predicting a process's future behavior based on historical event logs to support a process's execution. Many of the recent techniques utilize a machine-learned model to predict which event type is the next most likely. Beyond PBPM, prescriptive BPM aims at finding optimal actions based on considering relevant key performance indicators. Existing techniques are geared towards the outcome prediction and deal with alarms for interventions or interventions that do not represent process events. In this paper, we argue that the next event prediction is insufficient for practitioners. Accordingly, this research-in-progress paper proposes a technique for determining next best actions that represent process events. We conducted an intermediate evaluation to test the usefulness and the quality of our technique compared to the most frequently cited technique for predicting next events. The results show a higher usefulness for process participants than a next most likely event.

Keywords: Prescriptive business process monitoring, predictive business process monitoring, business process management, design science research.

1 Introduction

Predictive business process monitoring (PBPM) deals with the prediction of future behavior of running process instances [1] such as throughput times, next events or process outcomes [2]. A variety of PBPM techniques use machine learning (ML) algorithms [1, 2] which train predictive models based on historic process instances and subsequently use these models to make predictions about the most likely future process behavior of a running process instance. Performing the most likely behavior does not guarantee to achieve the desired business goal [3]. Two works [3, 4] have tackled this problem with prescriptive BPM techniques geared towards the outcome prediction task. Outcome prediction aims at classifying each ongoing process instance according to a given set of possible categorical outcomes, e.g., will the customer complain or not [5]. Transferred

15th International Conference on Wirtschaftsinformatik,
March 08-11, 2020, Potsdam, Germany

https://doi.org/10.30844/wi_2020_c12-weinzierl

to outcome-oriented prescriptive BPM, these categorical outcomes are alarms for interventions [3] or interventions [4]. An intervention itself does not represent a business process event but influences business process events to archive a desired key performance indicator (KPI). Beyond alarms and interventions, the next best actions that represent business process events can be recommended based on a defined KPI. For instance, the next most likely action after receiving a customer order would be the checking of the customer's credibility but aiming at high conversion of customers it might be better to immediately ship the ordered goods to the customer.

In our work, we address the determination of next best actions with our research goal: *providing and evaluating immediately a prescriptive BPM technique that recommends next best actions to optimize towards the KPI throughput time.* To reach this goal, we follow a design science research methodology [6] and develop an artifact in form of a prescriptive BPM technique as well as evaluate immediately its quality and usefulness with a real-life event log and a reputable baseline.

2 Related work on prescriptive business process monitoring

A variety of PBPM techniques were proposed by researchers as summarized by e.g., [1] or [2]. Many techniques maintain a focus on predicting a process instance's next event (e.g., [7] or [8]). Beyond PBPM, researchers recently advocated "prescriptive BPM" for addressing outcome-oriented prediction tasks. First, Teinemaa et al. [3] recognized the need for assessing predictions regarding their impact onto the process performance. Towards the first concept of a framework for alarm-based prescriptive BPM, they suggest a cost function for determining whether an intervention is reasonable or not. However, a concrete intervention is not determined. Further, they present a multi-perspective extension of their framework in [9]. Dees et al. [4] propose a prescriptive-oriented process-aware recommender system for risk-based outcome prediction. In addition to the work of Teinemaa et al. [3], this system creates interventions (i. e. emails) but these interventions don't show a preventive effect.

3 Towards a technique for determining next best actions

Our technique is geared towards the KPI throughput time and consists of a *learning* and a *recommendation phase*. The learning phase receives as input an event log $\mathcal{E}$ and outputs two ML models $\mathcal{M}$ learned on individually pre-processed versions of $\mathcal{E}$ (next event prediction and candidate selection). The recommendation phase based on the learning phase and realizes the prescriptiveness of the technique. Therefore, the four steps of the recommendation phase are detailed.

Predict suffix. Given a (newly) running process instance pi, the idea of this step is to predict the next sequence of events for each prefix of pi by a previously trained ML model $m_{sp} \in \mathcal{M}$ to calculate the throughput time of the complete process instance pi consisting of a prefix and a suffix afterwards. To train the model m_{sp} for predicting next events, we used the (deep) learning algorithm Long Short-term Neural Network

https://doi.org/10.30844/wi_2020_c12-weinzierl

(LSTM) [10] in the learning phase since LSTMs are recurrent neural networks designed to handle temporal dependencies in sequential problems [11], LSTMs can learn predictive models with a good predictive quality on different data sets [12], and finally, our baseline [8] use LSTMs, too. In sense of the suffix prediction, we repeatedly applied the model m_{sp} for next event prediction. For example, if the prefix of a process instance pi is $< A\{1\}, B\{1\} >$ at time step 2, the predicted suffix could be $< C\{1\}, D\{1\}, D\{1\}, E\{1\} >$. Here the value 1 assigned to each event is the value of a process-instance-based context attribute. Note the next event prediction starts at time step 2, since at time step 1 not enough information is available. Finally, the total throughput time for pi could be 10 hours, if A, B, C and E each take 1 hour, and D takes 3 hours.

Find candidates. In the second step, we want to find alternative (in our case "nearest") candidates for the predicted suffix. We consider this only necessary if the complete process instance (consisting of the prefix and the suffix) takes longer than the (KPI based) temporal threshold (average time over all process instances of $\mathcal{E}$). For instance, given the temporal threshold is 8 hours, the process instance pi at time step 2 is not in time since pi takes 10 hours. Therefore, we aim to find a better alternative for the suffix, in which we can define the first event for the next time step 3, to comply with the threshold. However, for selecting the suffix candidates, we used a second ML model $m_{cs} \in \mathcal{M}$. This model was trained with Ball Tree [13] ($k = 5$), which is a binary tree data structure for maintaining spatial data hierarchically and based on all possible suffixes that we can construct from the event log $\mathcal{E}$. We used a spatial-based technique for selecting the candidates to consider the semantic similarity between the suffixes of events and its assigned context information. So, for example, the model m_{cs} receives as input the predicted suffix $< C\{1\}, D\{1\}, D\{1\}, E\{1\} >$ of the process instance pi and outputs three similar suffixes. Note the number of similar suffixes is a hyperparameter of the Ball Tree model m_{cs}. In the best case, there exists a suffix such as $< G\{1\}, F\{1\}, E\{1\}>$ where the events F and E take 1 hour. So, the total throughput time of pi_{new} consisting of the prefix and the selected suffix is 6 hours and would be less than the given temporal threshold at 8 hours.

Select the best candidate. In the third step, the suffix with the lowest total throughput time is selected and the first event of this suffix is determined as the next best action.

Update of the running process instance. The next best action determined for time step 3 represents the last event of the new prefix $< A\{1\}, B\{1\}, G\{1\} >$ of pi_{new} in time step 3. The complete procedure is repeated until the process instance ends if the termination event is reached.

4 Preliminary results

To gain insight into the usefulness and quality of our approach, we developed a prototype for conducting an intermediate evaluation. For this, we used a sample of the bpi challenge 2019 event log [14]. We split the event log in a 2/3 training and 1/3 test set with a random process-instance-based sampling [15]. As a baseline we used the technique from Tax et al. [8] for calculating the probability of next events. First, we

https://doi.org/10.30844/wi_2020_c12-weinzierl

evaluated the usefulness through the percentage of process instances that could comply with the temporal threshold for different prefix lengths (cf., the left graph in **Figure 1**). Overall, our approach outperforms the baseline for most of the prefix lengths. Second, we evaluated the quality through the average Levenshtein distance which is a metric to determine the distance between two strings or sequences, i. e. the lower the value, the more similar two strings are [16] (cf., the right graph in **Figure 1**). With our approach this metric is generally lower, i.e. an updated process instance is more similar to the ground truth process instance for all prefix lengths.

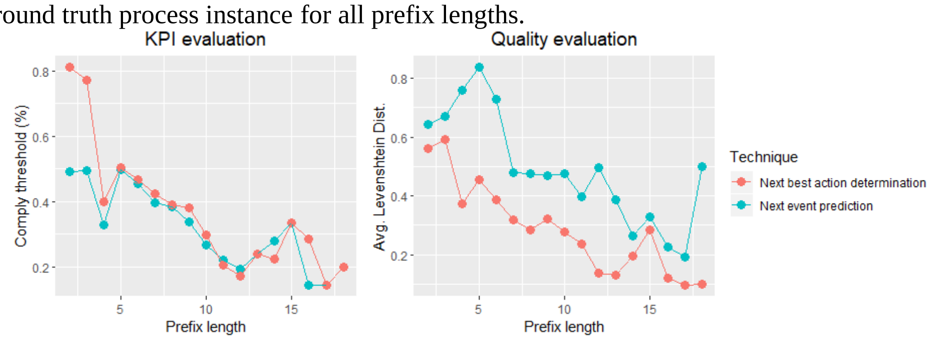

Figure 1. *Intermediate evaluation of our proposed approach*

5 Further research plans

Based on our identified research gap on prescriptive BPM, we argue that a crucial need for a prescriptive BPM technique for recommending next best actions in practice exists. With the development and intermediate evaluation of our artifact, we reached our research goal and could intermediately show that our technique can outperform the baseline regarding usefulness. That means, our technique can shorten the process throughput time and, maybe, reduce the risk to exceed contractual deadlines. In addition, the quality of our proposed technique is higher.

Further research will go in four major directions. First, we will investigate further methods to find semantically similar prefixes to improve the usefulness of our approach. Second, we will perform a profound evaluation including on the one hand more real-life event logs from which we can maybe directly derive the goal to be fulfilled. On the other hand, a domain-expert-based evaluation to find out if the events recommended as best action are really the best actions. Third, we plan to extend our technique by a concept-drift awareness and a root-cause analysis. With the presented technique, it is not possible to take events into account that have not been performed in the past or to a certain extend a proposed combination of actions that has not been performed in the past (or at least a similar one). To cope with that, a strategy for model retraining is necessary. Further, it is important to know which event of a process instance has a causal relationship with the defined threshold value triggering the next best action determination. A causality analysis can be used for that. Finally, we will create an abstract framework where our technique can be a possible instantiation. In general, there are different ways to design the learning and the recommendation phase or towards which single KPI or multiple KPIs the instantiation is tailored.

References

1. Márquez-Chamorro, A. E., Resinas, & M., Ruiz-Cortés, A.: Predictive monitoring of business processes: a survey. In: IEEE Transactions on Services Computing, vol. 11, pp. 962–977 (2017)
2. Di Francescomarino, C., Ghidini, C., Maggi, F. M., & Milani, F.: Predictive Process Monitoring Methods: Which One Suits Me Best? In: Proceedings of the 16th International Conference on Business Process Management, pp. 462–479. Springer, Cham (2018)
3. Teinemaa, I., Tax, N., de Leoni, M., Dumas, M., & Maggi, F. M.: Alarm-based prescriptive process monitoring. In: Proceedings of the 16th International Conference on Business Process Management, pp. 91-107. Springer, Cham (2018)
4. Dees, M., de Leoni, M., van der Aalst, W. M., & Reijers, H. A.: What if Process Predictions are not followed by Good Recommendations?. *arXiv preprint arXiv:1905.10173* (2019)
5. Teinemaa, I., Dumas, M., Rosa, M. L., & Maggi, F. M.: Outcome-oriented predictive process monitoring: In: ACM Transactions on Knowledge Discovery from Data, vol. 13 (2019)
6. Hevner, A. R., March, S. T., Park, J., & Ram, S.: Design Science in Information Systems Research. In: MIS Quarterly, vol. 28, pp. 75–105 (2004)
7. Breuker, D., Matzner, M., Delfmann, P., & Becker, J.: Comprehensible Predictive Models for Business Processes. In: MIS Quarterly, vol. 40, pp. 1009-1034 (2016)
8. Tax, N., Verenich, I., La Rosa, M., & Dumas, M.: Predictive Business Process Monitoring with LSTM Neural Networks. In: Proceedings of the 29th Conference on Advanced Information Systems Engineering, pp. 477-492. Springer, Cham (2017)
9. Fahrenkrog-Petersen, S. A., Tax, N., Teinemaa, I., Dumas, M., de Leoni, M., Maggi, F. M., & Weidlich, M.: Fire Now, Fire Later: Alarm-Based Systems for Prescriptive Process Monitoring. arXiv preprint arXiv:1905.09568 (2019)
10. Hochreiter, S., & Schmidhuber, J.: Long Short-Term Memory. In: Neural Computation, vol. 9, pp. 1735-1780 (1997)
11. LeCun, Y., Bengio, Y., & Hinton, G.: Deep learning. In: Nature, vol. 521, pp. 436-444 (2015)
12. Greff, K., Srivastava R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J.: LSTM: A Search Space Odyssey, In: IEEE Transactions on Neural Networks and Learning Systems, vol. 28, pp. 2222-2232 (2016)
13. Omohundro, S. M.: Five Balltree Construction Algorithms. Technical Report, International Computer Science Institute, Berkeley (1989)
14. van Dongen, B.F.: Dataset BPI Challenge 2019. 4TU. Centre for Research Data. https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1
15. Tama, B. A., & Comuzzi, M.: An empirical comparison of classification techniques for next event prediction using business process event logs. In: Expert Systems with Applications, vol. 129, pp. 233–245 (2019)
16. Yujian, L., & Bo, L.: A Normalized Levenshtein Distance Metric. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, pp. 1091-1095 (2007)

https://doi.org/10.30844/wi_2020_c12-weinzierl