Predictive business process monitoring with LSTMs

Tax, N.; Verenich, I.; La Rosa, M.; Dumas, M.

Published in:
Proceedings of the Twenty-Sixth Benelux Conference on Machine Learning (BENELEARN)

Published: 10/06/2017

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Predictive Business Process Monitoring with LSTMs

Niek Tax n.tax@tue.nl
Eindhoven University of Technology, The Netherlands

Ilya Verenich, Marcello La Rosa {ilya.verenich,m.larosa}@qut.edu.au
Queensland University of Technology, Australia

Marlon Dumas marlon.dumas@ut.ee
University of Tartu, Estonia

Keywords: deep learning, recurrent neural networks, process mining, business process monitoring

1. Introduction

Predictive business process monitoring techniques are concerned with predicting the evolution of running cases of a business process based on models extracted from historical event logs. A range of such techniques have been proposed for a variety of business process prediction tasks, e.g. predicting the next activity (Becker et al., 2014), predicting the future path (continuation) of a running case (Polato et al., 2016), predicting the remaining cycle time (Rogge-Solti & Weske, 2013), and predicting deadline violations (Metzger et al., 2015). Existing predictive process monitoring approaches are tailor-made for specific prediction tasks and not readily generalizable. Moreover, their relative accuracy varies significantly depending on the input dataset and the point in time when the prediction is made.

Long Short-Term Memory networks (Hochreiter & Schmidhuber, 1997) have been shown to deliver consistently high accuracy in several sequence modeling application domains, e.g. natural language processing and speech recognition. Recently, (Evermann et al., 2016) applied LSTMs specifically to predict the next activity in a case. This paper explores the application of LSTMs for three predictive business process monitoring tasks: (i) the next activity in a running case and its timestamp; (ii) the continuation of a case up to completion; and (iii) the remaining cycle time. The outlined LSTM architectures are empirically compared against tailor-made approaches using four real-life event logs.

2. Next Activity and Time Prediction

We start by predicting the next activity in a case and its timestamp. A log of business process executions consists of sequences (i.e., *traces*) of events, where for each event business task (i.e., *activities*) that was executed and the timestamp is known. Typically, the set of unique business tasks seen in a log is rather small, therefore learned representations (such as (Mikolov et al., 2013)) are unlikely to work well. We transform each event into a feature vector using a one-hot encoding on its activity.

If the last seen event occurred just before closing time of the company, it is likely that the next event of the trace will at earliest take place on the next business day. If this event occurred on a Friday and the company is closed during weekends, it is likely that the next event will take place at earliest on Monday. Therefore, the timestamp of the last seen activity is likely to be useful in predicting the timestamp of the next event. We extract two features representing the time domain: the time since the start of the business day, and the time since the start of the business week.

Figure 1 shows different setups that we explore. First, we explore predicting the next activity and its timestamp with two separate LSTM models. Secondly, we explore predicting them with one joint LSTM model. Thirdly, we explore an architecture with n-shared LSTM layers, followed by m task-specific layers. We use cross entropy loss for the predicted activity and mean absolute error (MAE) loss for the predicted time and train the neural network weights with Adam (Kingma & Ba, 2015).

For time prediction we take as baseline the set, bag, and sequence approach described in (van der Aalst et al., 2011). For activity prediction we take as baselines the LSTM based approach of (Evermann et al., 2016) and the technique of (Breuker et al., 2016). Ta-

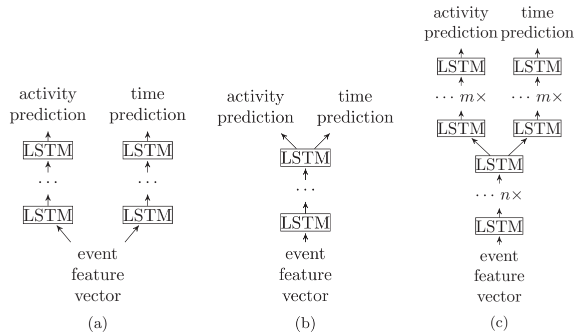

Figure 1. Neural Network architectures with single-task layers *(a)*, with shared multi-tasks layer *(b)*, and with $n+m$ layers of which n are shared *(c)*.

		Helpdesk		BPI'12 W	
Layers	Shared	MAE (time)	Accuracy (act.)	MAE (time)	Accuracy (act.)
3	3	3.77	0.7116	1.58	0.7507
3	2	3.80	0.7118	1.57	0.7512
3	1	3.76	**0.7123**	1.59	0.7525
3	0	3.82	0.6924	1.66	0.7506
2	2	3.81	0.7117	1.58	0.7556
2	1	3.77	0.7119	**1.56**	**0.7600**
2	0	3.86	0.6985	1.60	0.7537
1	1	**3.75**	0.7072	1.57	0.7486
1	0	3.87	0.7110	1.59	0.7431
		Time prediction baselines			
Set		5.83	-	1.97	-
Bag		5.74	-	1.92	-
Sequence		5.67	-	1.91	-
		Activity prediction baselines			
Evermann		-	-	-	0.623
Breuker		-	-	-	0.719

Table 1. Experimental results for the Helpdesk and BPI'12 W logs.

ble 1 shows the MAE of the predicted timestamp of the next event and the accuracy of the predicted activity on two data sets. It shows that LSTMs outperform the baseline techniques, and that architectures with shared layers outperform architectures without shared layers.

3. Suffix Prediction

By repeatedly predicting the next activity, using the method described in Section 2, the trace can be predicted completely until its end. The most recent method to predict an arbitrary number of events ahead is (Polato et al., 2016), which extracts a transition system from the log and then learns a machine learning model for each transition system state. Levenshtein similarity is a frequently used string similarity measure, which is based on the minimal number of insertions, deletions and substitutions needed to transform one string into another. In business processes, activities are frequently performed in parallel, leading to some event in the trace being arbitrarily ordered., therefore we consider it only a minor mistake when two events are predicted in the wrong order. We evaluate suffix predictions with Damerau-Levenshtein similarity, which adds a swapping operation to Levenshtein similarity. Table 2

Method	Helpdesk	BPI'12 W	Env. permit
(Polato et al., 2016)	0.2516	0.0458	0.0260
LSTM	**0.7669**	**0.3533**	**0.1522**

Table 2. Suffix prediction results in terms of Damerau-Levenshtein Similarity.

shows the results of suffix prediction on three data sets. The LSTM outperforms the baseline on all logs.

4. Remaining Cycle Time Prediction

By repeatedly predicting the next activity and its timestamp with the method described in Section 2, the timestamp of the last event of the trace can be predicted, which can be used to predict the remaining cycle time. Figure 2 shows the mean absolute error for each prefix size, for the four logs. As baseline we use the set, bag, and sequence approach described in (van der Aalst et al., 2011), and the approach described in (van Dongen et al., 2008). It can be seen that LSTM consistently outperforms the baselines for the Helpdesk log and the environmental permit log.

5. Conclusions

The foremost contribution of this paper is a technique to predict the next activity of a running case and its timestamp using LSTM neural networks. We showed that this technique outperforms existing baselines on real-life data sets. Additionally, we found that predicting the next activity and its timestamp via a single model (multi-task learning) yields a higher accuracy than predicting them using separate models. We then showed that this basic technique can be generalized to address two other predictive process monitoring problems: predicting the entire continuation of a running case and predicting the remaining cycle time.

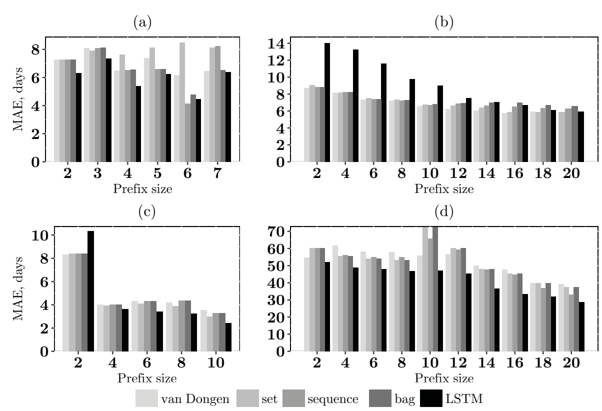

Figure 2. MAE values using prefixes of different lengths for *helpdesk* (a), *BPI'12 W* (b), *BPI'12 W (no duplicates)* (c) and *environmental permit* (d) datasets.

Acknowledgments

This work is accepted and to appear in the proceedings of the International Conference on Advanced Information Systems Engineering (Tax et al., 2017). The source code and supplementary material required to reproduce the experiments reported in this paper can be found at `http://verenich.github.io/ProcessSequencePrediction`. This research is funded by the Australian Research Council (grant DP150103356), the Estonian Research Council (grant IUT20-55) and the RISE_BPM project (H2020 Marie Curie Program, grant 645751).

References

Becker, J., Breuker, D., Delfmann, P., & Matzner, M. (2014). Designing and implementing a framework for event-based predictive modelling of business processes. *Proceedigns of the 6th International Workshop on Enterprise Modelling and Information Systems Architectures* (pp. 71–84). Springer.

Breuker, D., Matzner, M., Delfmann, P., & Becker, J. (2016). Comprehensible predictive models for business processes. *MIS Quarterly*, *40*, 1009–1034.

Evermann, J., Rehse, J.-R., & Fettke, P. (2016). A deep learning approach for predicting process behaviour at runtime. *Proceedings of the 1st International Workshop on Runtime Analysis of Process-Aware Information Systems*. Springer.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.

Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference for Learning Representations*.

Metzger, A., Leitner, P., Ivanovic, D., Schmieders, E., Franklin, R., Carro, M., Dustdar, S., & Pohl, K. (2015). Comparing and combining predictive business process monitoring techniques. *IEEE Trans. Systems, Man, and Cybernetics: Systems*, *45*, 276–290.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems* (pp. 3111–3119).

Polato, M., Sperduti, A., Burattin, A., & de Leoni, M. (2016). Time and activity sequence prediction of business process instances. *arXiv preprint arXiv:1602.07566*.

Rogge-Solti, A., & Weske, M. (2013). Prediction of remaining service execution time using stochastic Petri nets with arbitrary firing delays. *Proceedings of the International Conference on Service Oriented Computing* (pp. 389–403). Springer.

Tax, N., Verenich, I., La Rosa, M., & Dumas, M. (2017). Predictive business process monitoring with LSTM neural networks. *Proceedings of the International Conference on Advanced Information Systems Engineering* (p. To appear.). Springer.

van der Aalst, W. M. P., Schonenberg, M. H., & Song, M. (2011). Time prediction based on process mining. *Information Systems*, *36*, 450–475.

van Dongen, B. F., Crooy, R. A., & van der Aalst, W. M. P. (2008). Cycle time prediction: when will this case finally be finished? *Proceedings of the International Conference on Cooperative Information Systems* (pp. 319–336). Springer.