Felix Wolff

Exposé: Deep Learning Next-Activity Prediction With Cluster-Based Input Data

Abstract – Predictive process monitoring is a relatively recent application of predictive analytics in the context of business processes. It is concerned with anticipating the future behaviour of running process instances, based on logs of completed instances. In this thesis we aim at forecasting the next activity in a running case based on event history and data attribute history. The forecasts shall be made via LSTM neural networks trained on clustered input data. One network shall be trained per cluster, while the encoding of past events in the input data shall also be engineered. Performance benchmarks against related recent publications shall complete the thesis.

Motivation

The increasing numbers of knowledge workers brought with it the advent of adaptive case management (ACM) to support them [Dru99; LM09]. As knowledge workers do not follow a clear and structured approach in their data-intensive work, supporting them via traditional means of business process management is not the best option. One traditional approach would be the discovery of process models from process logs to analyze and optimize the workflow. Due to the unstructured nature of knowledge work, an attempt at doing so would likely result in a large *spaghetti model* [Aal16, ch.14].

The course that a case can take is highly dependent on a number of environmental factors, such as data used and produces inside a case [Sch+18], and as such can be hard to foresee. Numerous applications of predictive analytics have proven great accuracy in this context. These applications are attributed to the domain of predictive process monitoring. Mostly variables such as remaining cycle time and case outcome have been targeted, but few attempts have been made to target a sequence or the next activity [Fra+18; RSW13]. This leaves open an opportunity to assist knowledge workers in their daily work without the need for discovering a process model [Hau+14]. In the long run, one could imagine a recommendation system targeting optimal case outcomes with respect to the individual case state [LM09; Hau+14].

Background

Predictive process monitoring: Process science revolves around managing and optimizing structured procedures, while the broad area of data science covers data mining, algorithmic analysis and predictive analytics. Bridging the gap between the two fields is process mining [Aal16, p.18]. It covers the three steps of model discovery, conformance checking and model enhancement [Aal16].

These three steps are focused on offline data. If one would like to avoid a certain process outcome or e.g. an SLA violation, a guess at future developments requires resorting to online data analysis. At this step, techniques from the domain of predictive analytics can be employed[1].

Predictive analytics brings together a variety of statistical techniques like data mining, predictive modelling, and machine learning in order to make predictions about future events throught the use of historical data. In the domain of business processes, statistical or machine learning models are trained with historical process execution logs and *target* a specific piece of data that should be predicted. This application is called predictive process monitoring and allows answering questions such as *Given the current state of things, will I still meet my SLA?* or *Given the current case state, how long is this case still going to take?*. The answers to such questions can give case managers the opportunity to intervene if a case takes an unwanted course or might fail to meet KPI requirements.

Model training: Predictive analytics is a lot about model training, and the rough outline of necessary steps necessary to train a model is listed here:

1. Determine the *target* variable, it is the variable that is supposed to be predicted

2. Preprocess the dataset. This can mean introducing one-hot encodings, normalized values, but also basic data quality assurance such as null value elimination. Feature engineering can also happen at this step.

3. Partition the dataset into two parts. One part is set aside for model performance verification, as there the actual target variable value is known. This is commonly referred to as the *test set*, while the remainder is called the *training set*.

[1] More detail from Marlon Dumas on how these topics fit together: `https://www.youtube.com/watch?v=hMQolsRTOKO`

4. Train the model on the training set. Models are trained multiple times with different hyper-parameters to find the optimum configuration with respect to prediction accuracy on the test set. Hyper-parameters are model-specific values such as cutoff-thresholds that have impact on model performance.

Neural networks with long short-term memory: An artificial neural network is an example for a machine learning model. It is made up of neurons, similar to its organic counterpart. The network is organized in three types of layers: a single input layer, one or more hidden layers and a single output layer. The neurons (being mathematical functions), pass their output on to those in the next layer via weighted connections. The weights on these connections are changed as the network is trained [Ros58]. Improving this forward-feeding network with backpropagation, i.e. learning from errors, made applications on pattern-detection successful[2]. Finally, enhancing the network with a way to remember sequences of events allows application on time-series data. The capacity as well as the durability of this memory are purposely limited as to avoid overfitting. As such, a long short-term memory inside a neural network functions similarly to our human one: we can remember a certain number of things for a short time, but we do forget some of them. The LSTM feature also equips the network with a remember and a forget capacity [HS97].

Ensemble learning: Predictive models vary in complexity, and with it varies their amount of bias and variance. It can be said that more complex models (such as neural networks or random forests) deliver higher variance and lower bias in their predictions, whereas simpler models (like regressions) exhibit lower variance and higher bias [Les15]. This is commonly referred to as the bias-variance tradeoff[3]. Using several complex models as a basis and training another model using their predictions leverages the diverse strengths of the base models to cancel out their weaknesses [Tso+09]. See figure 0.1 for an illustration. Diversity at the base of such an ensemble can, as so often in life, be the key to good outcomes. The ensemble strategy (often also referred to as model stacking) should be as simple as possible to avoid further complex rounds of hyper-parameter tuning.

Related Work

Hauder et al. mention numerous research challenges in the domain of ACM, among them an active support system for knowledge workers [Hau+14]. The need for such

[2]Backpropagation can be attributed to many authors, as Schmidhuber blogs: `http://people.idsia.ch/~juergen/who-invented-backpropagation.html`

[3]Wikipedia has an exhaustive definition: `https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff`

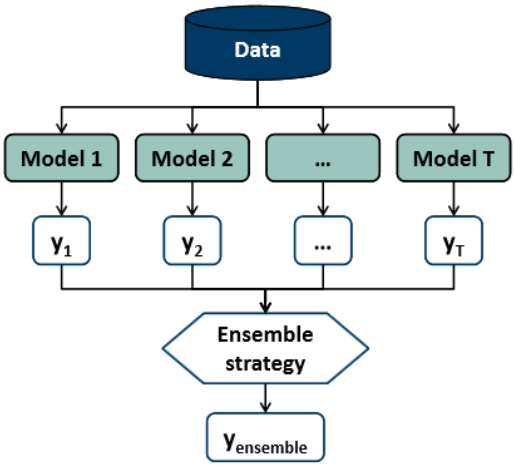

Figure 0.1: A generic ensemble architecture, taken from lecture material [Les15]

a system is emphasized by Francescomarino et al. in their literature review, where it has been found that few prediction approaches target the next activity [Fra+18].

An example for how such a system might look like is given by Huber, who has developed a next-step recommendation system serving different case goals. The system is prototypically implemented into CoCaMa[4], a prototypical case management application. The system has been evaluated with 25 hand-made case logs.

Building upon each other are the works by Evermann et al. [Eve+16] and Schönig et al. [Sch+18]. Evermann et al. have successfully demonstrated the good performance of long-short-term memory (LSTM) neural networks in predicting the next activity. Their approach did not take into account specific case data attributes however. How making use of this contextual information can improve the prediction accuracy even more, has been shown by Schönig et al. [Sch+18]. Furthermore Schönig et al. have explored data preparation methods for supporting the model during learning.

Similarly, Polato et al. make use of environmental information in their work for improving the prediction of the remaining time of business process instances [Pol+14].

Metzger et al. predict run-time of a case by comparing and combining different prediction models into a model ensemble. Then, the members of the ensemble are selected based on their predictive performance measures. This allows taking into account costs of false predictions [Met+15].

[4]CoCaMa is an abbreviation for a project called Collaborative Case Management, which appears to be retired: `http://archive.li/uZFnN`

Francescomarino et al. have performed clustering in the preprocessing phase of model training and prediction. Having clustered the training data, one model was created and trained for each cluster. For obtaining a prediction, the optimal cluster for a new data item is found from which the corresponding model is selected. This approach was evaluated on the accuracy of predicate fulfillment with two different clustering methods (k-means and DBSCAN) and two different prediction models (decision trees and random forests) [Fra+15]. A further evaluation criteria was *earliness*, i.e. at which point in time the correct result could be determined.

Thesis objective

In my thesis, I want to investigate the synergies of combining the aforementioned approach of Francescomarino et al. for learning data clustering with LSTM neural networks as per Evermann et al. Case data attributes shall be used during model training and prediction, as Polato et al. [Pol+14] and Schönig et al. [Sch+18] demonstrated their usefulness.

This would contribute to a field of research which is currently being explored and where LSTM networks have been applied successfully on prediction problems with long-term dependencies [Eve+16; Tax+17; Sch+18; GS05].

Furthermore, I want to determine how historical case log data is prepared best for learning, as only Schönig et al. has written a small subsection on this [Sch+18]. If time permits, I also want to investigate the potential of ensembles within this context, as they can potentially enlighten the user about the reason for a prediction. With neural networks it is hard to comprehend the reasons behind a prediction. Other types of models deliver better comprehensibility.

Throughout the document I will strive to meet recently demanded machine learning paper quality criteria [LS18].

The performance of the combined approaches shall be evaluated against the data from the Business Process Intelligence Challenges (BPIC) 2011, 2012 and 2017 [Bpia; Bpib; Bpic]. This allows for comparison with the results of Francescomarino et al., Evermann et al., Tax et al. and Schönig et al. [Fra+18; Eve+16; Tax+17; Sch+18]. The next steps and an approximate timeframe are shown in the table below:

Date	Milestone
September	Hardware procurement; Reproduction of the results of Schönig et al.
October	Clustering of data via Francescomarinos method
November	Hyper-parameter tuning of LSTM neural network on clustered training data
December	Training and benchmarking architecture setup
January	Benchmarking
February	Thesis hand-in
March 1st	Defense

Table 0.1: Approximate timeframe for my work

Bibliography

[Aal16] Wil M. P. van der Aalst. *Process Mining: Data Science in Action*. 2nd ed. Heidelberg: Springer, 2016 (cit. on pp. 1, 2).

[Bpia] *BPI Challenge 2011*. `https://data.4tu.nl/repository/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54`. Accessed: 2018-08-20 (cit. on p. 5).

[Bpib] *BPI Challenge 2012*. `https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f`. Accessed: 2018-08-15 (cit. on p. 5).

[Bpic] *BPI Challenge 2017*. `https://data.4tu.nl/repository/uuid:7e326e7e-8b93-4701-8860-71213edf0fbe`. Accessed: 2018-08-16 (cit. on p. 5).

[Dru99] Peter F. Drucker. „Knowledge-Worker Productivity: The Biggest Challenge“. In: *California Management Review* 41.2 (1999), pp. 79–94. eprint: `https://doi.org/10.2307/41165987` (cit. on p. 1).

[Eve+16] Joerg Evermann, Jana-Rebecca Rehse, and Peter Fettke. „A Deep Learning Approach for Predicting Process Behaviour at Runtime“. In: *Business Process Management Workshops*. 2016 (cit. on pp. 4, 5).

[Fra+15] Chiara Di Francescomarino, Marlon Dumas, Fabrizio Maria Maggi, and Irene Teinemaa. „Clustering-Based Predictive Process Monitoring“. In: *CoRR* abs/1506.01428 (2015) (cit. on p. 5).

[Fra+18] Chiara Di Francescomarino, Chiara Ghidini, Fabrizio Maria Maggi, and Fredrik Milani. „Predictive Process Monitoring Methods: Which One Suits Me Best?“ In: *CoRR* abs/1804.02422 (2018). arXiv: `1804.02422` (cit. on pp. 1, 4, 5).

[GS05] Alex Graves and Juergen Schmidhuber. „Framewise phoneme classification with bidirectional LSTM networks“. In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* 4 (2005), 2047–2052 vol. 4 (cit. on p. 5).

[Hau+14] Matheus Hauder, Simon Pigat, and Florian Matthes. „Research Challenges in Adaptive Case Management: A Literature Review“. In: *2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations* (2014), pp. 98–107 (cit. on pp. 1, 3).

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. „Long Short-Term Memory“. In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780 (cit. on p. 3).

[Les15] Stefan Lessmann. *Business Analytics & Data Science lecture*. 2015/2016 (cit. on pp. 3, 4).

[LM09] Craig LeClair and Connie Moore. „Dynamic Case Management — An Old Idea Catches New Fire“. In: (2009) (cit. on p. 1).

[LS18] Z. C. Lipton and J. Steinhardt. „Troubling Trends in Machine Learning Scholarship“. In: *ArXiv e-prints* (July 2018). arXiv: 1807.03341 [stat.ML] (cit. on p. 5).

[Met+15] A. Metzger, P. Leitner, D. Ivanović, et al. „Comparing and Combining Predictive Business Process Monitoring Techniques“. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45.2 (2015), pp. 276–290 (cit. on p. 4).

[Pol+14] Mirko Polato, Alessandro Sperduti, Andrea Burattin, and Massimiliano de Leoni. „Data-aware remaining time prediction of business process instances“. In: *2014 International Joint Conference on Neural Networks (IJCNN)* (2014), pp. 816–823 (cit. on pp. 4, 5).

[Ros58] F. Rosenblatt. „The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain“. In: *Psychological Review* (1958), pp. 65–386 (cit. on p. 3).

[RSW13] Andreas Rogge-Solti and Mathias Weske. „Prediction of Remaining Service Execution Time Using Stochastic Petri Nets with Arbitrary Firing Delays“. In: *ICSOC*. 2013 (cit. on p. 1).

[Sch+18] Stefan Schönig, Richard Jasinski, Lars Ackermann, and Stefan Jablonski. „Deep Learning Process Prediction with Discrete and Continuous Data Features“. In: *Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering*. s.l., 2018 (cit. on pp. 1, 4, 5).

[Tax+17] Niek Tax, Ilya Verenich, Marcello La Rosa, and Marlon Dumas. „Predictive Business Process Monitoring with LSTM Neural Networks“. In: *CAiSE*. 2017 (cit. on p. 5).

[Tso+09] Grigorios Tsoumakas, Ioannis Partalas, and Ioannis P. Vlahavas. „An Ensemble Pruning Primer“. In: *Applications of Supervised and Unsupervised Ensemble Methods*. 2009 (cit. on p. 3).