Leveraging Multi-view Deep Learning for Next Activity Prediction*

Vincenzo Pasquadibisceglie[1], Annalisa Appice[1,2], Giovanna Castellano[1,2], and Donato Malerba[1,2]

[1] Department of Informatics, Università degli Studi di Bari Aldo Moro, via Orabona, 4 - 70125 Bari - Italy
vincenzo.pasquadibisceglie@uniba.it,annalisa.appice@uniba.it, giovanna.castellano@uniba.it,donato.malerba@uniba.it
[2] Consorzio Interuniversitario Nazionale per l'Informatica - CINI, Italy
Tel.: +39-080-5443262

Abstract. Predicting the next activity in a running trace is a fundamental problem in business process monitoring since such predictive information may allow analysts to intervene proactively and prevent undesired behaviors. This paper describes a predictive process approach that couples multi-view learning and deep learning, in order to gain accuracy by accounting for the variety of information possibly recorded in event logs. Experiments with benchmark event logs show the accuracy of the proposed approach compared to several recent state-of-the-art methods.

Keywords: Predictive process mining · Next actvitiy prediction · Deep Learning · Multi-view Learning

1 Introduction

Nowadays predictive process mining is playing a fundamental role in the business scenario as it is emerging as an effective means to monitor the execution of any business running process. In particular, knowing in advance the next activity of a running process instance may foster an optimal management of resources and promptly trigger remedial operations to be carried out. Recently, accounting for the results achieved with deep artificial neural networks, significant interest has arisen in applying deep learning to analyze event logs and gain accurate insights into the future activities of the logged processes (e.g. [1,5,6,8,9]). However, the common approach in these studies is to simply consider an event from the single perspective of the executed activities with their timestamps. Based on these premises, we have recently proposed a richer representation that takes into account different perspectives for each trace. In particular, in [7], we have introduced a process predictive approach called MiDA (Multi vIew Deep learning based approach for next Activity prediction) for yielding accurate prediction of the next activity in a running trace.[3] MiDA combines multi-view learning with

[3] The source code of the proposed approach is available on the GitHub repository https://github.com/vinspdb/MiDA

deep learning. Specifically, it resorts to a multi-view input scheme that injects each characteristic-based view of an event into a deep neural network with Long Short-Term Memory (LSTM) layers. These layers are able to process the multi-view information by taking into account the sequential nature of event logs in business processes. In short, the advantage of our proposal is that the information collected along any process perspective can be, in principle, taken into account to gain predictive accuracy. Experiments with various benchmark event logs show the accuracy of the proposed approach compared to several recent state-of-the-art methods. The paper is organized as follows. Section 2 reports preliminary concepts, while Section 3 describes the proposed approach. In Section 4 we describe the experimental setting and discuss the relevant results. Finally, Section 5 draws conclusions.

2 Preliminary concepts

The basic assumption is that the event log contains information on activities executed for specific traces of a certain process type, as well as their durations and any other optional characteristics (e.g. resources, costs). So, an *event* e is a complex entity characterized by a set of mandatory characteristics, that are the activity and its timestamp indicating date and time of occurrence calculated as the time elapsed from the start of the event. In addition, an event may be associated with a set of optional characteristics, such as the resource triggering the activity, the life cycle of the activity or the cost of completing the activity. An event log is a set of events. Each event in the log is linked to a trace and is globally unique. A *trace* σ represents the execution of a process instance. It is a finite sequence of distinct events, such that time is non-decreasing in the trace (i.e. for $1 \leq i < j \leq |\sigma|$: $e_i.Timestamp \leq e_j.Timestamp$ with $|\sigma| = length(\sigma)$). An *event log* $\mathcal{L} = \{\sigma_i\}_{i=1}^{N}$ is a bag of N traces. By accounting for the structure of events, traces of an event log can be characterized by different views. A *view* is a description of the traces along a specific event characteristic (perspective). Therefore, every event log can be defined on the mandatory views that are associated with the activities and the timestamps, as well as on additional views associated with the optional characteristics of events. A *prefix trace* $\sigma_i^k = \langle e_1, e_2, ..., e_k \rangle$ is a sub-sequence of a trace starting from the beginning of the trace. Of course from each trace σ we can derive several prefix traces σ^k with $1 \leq k = |\sigma^k| \leq |\sigma|$. Hence, a trace is a complete process instance (started and ended), while a prefix trace is an instance in execution (running trace).

3 MiDA

In MiDA, each prefix trace is represented on every mandatory perspective recorded in the log (activities and timestamps), as well as on every additional perspective possibly recorded in the log (e.g. resource, life cycle, cost). In particular, we consider both categorical attributes (e.g. the activities and the resources) and numerical attributes (e.g. the timestamp) to represent each event. Hence

given a prefix trace $\sigma_i^k = \langle e_{i1}, e_{i2}, ..., e_{ik} \rangle$, each event $e_{ij} \in \sigma_i^k$ is defined by both categorical and numerical attributes. We indicate by $\mathbf{A}_C$ the set of categorical attributes and by $\mathbf{A}_N$ the set of numerical attributes characterizing an event. A padding technique is adopted to deal with equal-length prefix traces with length equal to AVG_l (average trace length).

Every categorical attribute in $\mathbf{A}_C$ is converted into a numerical representation, in order to be processed by a neural network. For each categorical attribute $att_l \in \mathbf{A}_C$ having vocabulary $\mathcal{V}_l$, we define a coding function:

$$f : \mathcal{V}_l \rightarrow [0, 1, 2, ..., |\mathcal{V}_l|],$$

that univocally assigns an integer value to each categorical value in the vocabulary $\mathcal{V}_l$ of attribute att_l. On the other had, every numerical attribute in $\mathbf{A}_N$, such as those related to temporal information of the events, does not require any coding since their real values can be directly processed by the neural network.

However, the structured integer representation for categorical attributes introduced above is not directly applicable to be processed by a neural network, due to the continuous nature of neural computation. So, to treat both integer-valued views and real-valued views of traces in a unified manner, we use the entity embedding method [3] to automatically learn a multi-dimensional real-valued representation of categorical views. Given the integer-valued representation of a categorical view $\mathbf{x} = (x_1, x_2, ..., x_{AVG_L})$ we fed it into an extra layer of linear neurons, called embedding layer, that maps each integer value in x_i to an *entity embedding*, i.e. a fixed size vector $\mathbf{y}_i \in \mathbb{R}^d$. Hence a 1D integer-valued vector $\mathbf{x}$ (size AVG_L) is mapped to a 2D real-valued matrix Y (size $d \times AVG_L$). The matrix Y, called embedding matrix, is jointly learned with the model during training of the neural network. The size d of an embedding layer is $d = [D/2]$, where D is the cardinality of the vocabulary $\mathcal{V}$. Then the output of the embedding layers are concatenated into a single vector that represents a high-level representation of the multiple views information related to events in traces. This high-level representation is fed into a recurrent neural network module composed of two stacked LSTM layers. The first LSTM layer provides a sequence output to fed the second LSTM layer.

The LSTM approach is used as the core of our deep learning architecture since it is suitable to process sequences, such as those underlying a business process event log. The LSTM recurrent module used in our deep architecture also includes two Batch Normalization layers that are interspersed with the two LSTM layers, in order to accelerate the learning process. Finally, the output of the LSTM module is fed into a softmax layer, in order to compute the final output (i.e. the next activity) from probabilities of different classes (activities) computed using the *softmax* activation function. The training of the network is accomplished by the Backpropagation algorithm that has been applied with early stopping to avoid overfitting. In particular, the training phase is stopped when there is no improvement of the loss on the validation set for 20 consecutive epochs. To minimize the loss function we use the Nadam optimizer. The maximum number of epochs was set to 200. The optimization phase of the hyper-parameters (learning rate in [0.00001, 0.01], LSTM unit size among 50, 75 and

Table 1: Event log description

Event log	♯Traces	♯Events	♯Activities	Perspectives
BPI12	13087	164506	23	activity, timestamp, resource, loan amount
BPI13Incident	7554	65533	13	activity, timestamp, resource, impact, org group, org role, org country, org involved, product, resource country
BPI13Problem	2306	9011	7	activity, timestamp, resource, impact, org group, org role, org country, org involved, product, resource country
Receipt	1434	8577	27	activity, timestamp, resource, channel, department, group, org group, responsible
BPI17Offer	42995	193849	8	activity, timestamp, resource, monthly cost, credit score, first withdrawal amount, offered amount, number of terms, action
BPI20Request	6886	36796	19	activity, timestamp, resource, org, project, task, role

Table 2: Characteristics of the compared methods

Method	Perspectives	Embedding	Deep Learning architecture
MiDA	all	Yes	LSTM
[6]	activity, timestamp, resource	No	CNN
[1]	activity, timestamp, role	Yes	LSTM
[9]	activity, timestamp	No	LSTM
[2]	activity, resource	Yes	LSTM
[5]	activity, timestamp	No	CNN

100, and Batch size in $[2^5, 2^{10}]$) is conducted by using the 20% of the training set as validation set and performing optimization with SMAC [4]. The cross-entropy loss function is used for optimization.

4 Experiments

To provide a compelling evaluation of the effectiveness of our approach, we have conducted a range of experiments on eighth benchmark event logs[4]. Table 1 summarizes the characteristics of the considered logs. The main objective of these experiments is to investigate the performance of MiDA compared to that of the most recent state-of-the-art deep learning methods that address the task of predicting the next activity of a running trace. We compare our method to that of [6],[1], [9], [2] and [5]. Table 2 reports the characteristics of the compared methods. We run the state-of-the-art methods using the sets of hyper-parameters considered in the reference studies. The source codes of these approaches are

[4] The event logs are available on https://data.4tu.nl

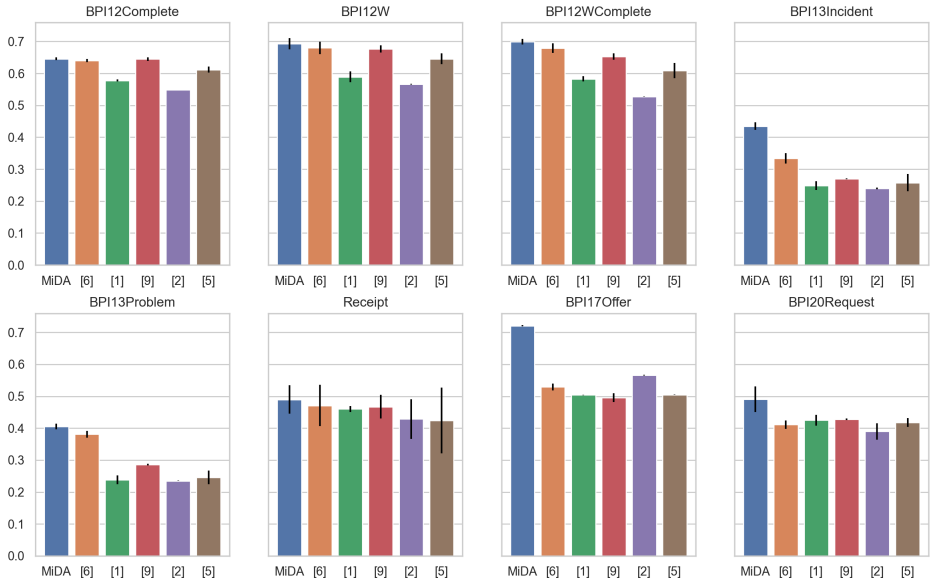

Fig. 1: Comparison between MiDA and related methods defined by in [6], [1], [9], [2] [5] in terms of Fscore Mean and standard deviation of metrics are reported.

publicly available. Hence, we evaluate all the methods on the same event log splits. In particular, for each event log, we evaluate the performance of each compared approach by partitioning the event log in training and testing traces according to a 3-fold cross validation.

Figure 1 collects the Fscore metric of MiDA and the baselines. These results provide the empirical evidence that MiDA is systematically more accurate than the evaluated baselines along Fscore metric.

5 Conclusion

In this paper, we have illustrated a novel multi-input, deep learning-based, business process predictive approach recently proposed in [7]. This approach can take advantage of all the characteristics possibly recorded with events. In particular, we couple a multi-view learning approach with a deep learning architecture, in order to gain predictive accuracy from the diversity of data in each view without suffering from the curse of dimensionality. The experiments performed on several event logs confirm the effectiveness of the proposed approach.

One limitation of the proposed approach is the lack of prescription and explanation with predictions. A research direction is that of enriching traditional business process mining approaches, that are able to discover interpretable models of processes, with the predictive ability of a deep learning architectures and

take advantage of the interpretability of the model for the prescriptive scope. Additional directions for further work include the extension of the proposed approach to deal with the presence of a condition of activity imbalance, i.e. activities that occur less frequently, in an event log. For example, techniques of training data augmentation may be explored, in order to achieve the balanced condition in the learning stage.

Acknowledgments

The research of Vincenzo Pasquadibisceglie is funded by PON RI 2014-2020 - Big Data Analytics for Process Improvement in Organizational Development - CUP H94F18000270006.

References

1. Camargo, M., Dumas, M., Rojas, O.G.: Learning accurate LSTM models of business processes. In: Hildebrandt, T.T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) International Conference on Business Process Management, BPM 2019. LNCS, vol. 11675, pp. 286–302. Springer (2019). https://doi.org/10.1007/978-3-030-26619-6_19
2. Evermann, J., Rehse, J.R., Fettke, P.: Predicting process behaviour using deep learning. Decision Support Systems **100**, 129 – 140 (2017). https://doi.org/https://doi.org/10.1016/j.dss.2017.04.003, smart Business Process Management
3. Guo, C., Berkhahn, F.: Entity embeddings of categorical variables. CoRR **abs/1604.06737** (2016), `http://arxiv.org/abs/1604.06737`
4. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Coello, C.A.C. (ed.) Learning and Intelligent Optimization - 5th International Conference, Selected Papers. LNCS, vol. 6683, pp. 507–523. Springer (2011). https://doi.org/10.1007/978-3-642-25566-3_40
5. Pasquadibisceglie, V., Appice, A., Castellano, G., Malerba, D.: Using convolutional neural networks for predictive process analytics. In: International Conference on Process Mining, ICPM 2019. pp. 129–136 (2019). https://doi.org/10.1109/ICPM.2019.00028
6. Pasquadibisceglie, V., Appice, A., Castellano, G., Malerba, D.: Predictive process mining meets computer vision. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) Business Process Management Forum, BPM 2020. pp. 176–192. Springer International Publishing, Cham (2020)
7. Pasquadibisceglie, V., Appice, A., Castellano, G., Malerba, D.: A multi-view deep learning approach for predictive business process monitoring. IEEE Transactions on Services Computing pp. 1–1 (2021). https://doi.org/10.1109/TSC.2021.3051771
8. Rama-Maneiro, E., Vidal, J.C., Lama, M.: Deep learning for predictive business process monitoring: Review and benchmark (2021)
9. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: Dubois, E., Pohl, K. (eds.) International Conference on Advanced Information Systems Engineering, CAISE 2017. pp. 477–492. Springer (2017)