Expert Systems With Applications 87 (2017) 1–14

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Run-time prediction of business process indicators using evolutionary decision rules

Alfonso E. Márquez-Chamorro*, Manuel Resinas, Antonio Ruiz-Cortés, Miguel Toro

Dpto. Lenguajes y Sistemas Informáticos, University of Seville, Seville, Spain

ARTICLE INFO

Article history:
Received 10 March 2017
Revised 9 May 2017
Accepted 27 May 2017

Keywords:
Business process management
Process mining
Predictive monitoring
Business process indicator
Evolutionary algorithm

ABSTRACT

Predictive monitoring of business processes is a challenging topic of process mining which is concerned with the prediction of process indicators of running process instances. The main value of predictive monitoring is to provide information in order to take proactive and corrective actions to improve process performance and mitigate risks in real time. In this paper, we present an approach for predictive monitoring based on the use of evolutionary algorithms. Our method provides a novel event window-based encoding and generates a set of decision rules for the run-time prediction of process indicators according to event log properties. These rules can be interpreted by users to extract further insight of the business processes while keeping a high level of accuracy. Furthermore, a full software stack consisting of a tool to support the training phase and a framework that enables the integration of run-time predictions with business process management systems, has been developed. Obtained results show the validity of our proposal for two large real-life datasets: BPI Challenge 2013 and IT Department of Andalusian Health Service (SAS).

1. Introduction

Process mining techniques allow the extraction of useful information from the event log and historical data of business processes (Li et al., 2016a; Schnig, Cabanillas, Jablonski, & Mendling, 2016). Knowledge can be generated from this information to improve the processes (Kamsu-Foguem, Rigal, & Mauget, 2013; Nkambou, Fournier-Viger, & Mephu-Nguifo, 2011; Potes-Ruiz, Kamsu-Foguem, & Grabot, 2014). Generally, this knowledge is extracted after the process has been finished. Nevertheless, the interest to apply process mining to running process instances is increasing (Maggi, Francescomarino, Dumas, & Ghidini, 2014). One of the main issues in process mining is the predictive monitoring of business processes (de Leoni, van der Aalst, & Dees, 2016). The main value of predictive monitoring is to provide information in order to take proactive and corrective actions to improve process performance and mitigate risks in real time. Predictive monitoring of business process provides the prediction of business process indicators of a running process instance with the generation of predictive models. Business process indicators are quantifiable metrics that can be measured directly by data that is generated within the process flow (del Río-Ortega, Resinas, Cabanillas, & Ruiz-Cortés, 2013). An improvement in the prediction of these indicators, in many occasions, also means savings in human and economic resources and prevention of important loss of turnover to the companies. Some issues of real companies can also be solved with predictive monitoring. For instance, Push to front problem, detailed in Verbeek (2013) and covered in this work, try to identify those incidences which are not resolved by the service desks and are pushed to the other support lines of the company.

Since predicting these process indicators can be interpreted as a classification or regression problem, machine learning algorithms can be used for this task (Francescomarino, Dumas, Maggi, & Teinemaa, 2015; Maggi, Francescomarino, Dumas, & Ghidini, 2014). Classification and regression are used for the prediction of discrete or continuous target values, respectively. For instance, an indicator such as, the cycle time of a process instance can be regarded as a regression problem. By contrast, the fulfillment of a determined target, *e.g.* the process instance must complete in less than 4 h, or a condition, *e.g.* whether a specific activity occurs in the process instance, can be interpreted as a classification problem.

Multiple machine learning approaches have been applied for predictive monitoring, such as decision trees (Maggi, Francescomarino, Dumas, & Ghidini, 2014), clustering methods (Francescomarino, Dumas, Maggi, & Teinemaa, 2015) or neural networks (Tax, Verenich, Rosa, & Dumas, 2016). Nevertheless, as far as we are concerned, evolutionary algorithms (EAs) have not

* Corresponding author.
E-mail addresses: amarquez6@us.es, alfedu@gmail.com (A.E. Márquez-Chamorro), resinas@us.es (M. Resinas), aruiz@us.es (A. Ruiz-Cortés), migueltoro@us.es (M. Toro).

http://dx.doi.org/10.1016/j.eswa.2017.05.069
0957-4174/

been applied for the prediction of process indicators. The use of an evolutionary algorithm may be justified for four different reasons (Fogel, 1997): *(a)* it can handle continuous and discrete attributes and automatically discretizes the continuous features; *(b)* it also handle missing attribute values and noise; *(c)* it can build models that can be easily interpreted by humans and finally *(d)* it finds a sub-set of the features that are relevant to the classification without the use of feature selection. In addition, EAs have shown the capacity of finding suboptimal solutions in search spaces when the search space is characterized by high dimensionality (Marquez-Chamorro, Asencio-Cortes, Divina, & Aguilar-Ruiz, 2014). In this case, the set of possible state conditions of a process, encoding in decision rules, determine the search space and fulfil these requirements. Some methods in process mining area also utilize association or decision rules for the improvement of the performance of the processes (Karray, Chebel-Morello, & Zerhouni, 2014; Wen, Zhong, & Wang, 2015).

In this work, we have developed a general method based on an evolutionary rule learning approach for the prediction of business process indicators in execution time. The resulting model consists in a set of decision rules that determine a prediction for an indicator of a running process instance. We have employed as encoded features, a window of the previous events to the point in the process execution where the prediction is carried out. This window of events considers attributes of a typical event log, such as activity name or timestamps, together with the data of each event. A combination of continuous and discrete values is allowed by the evolutionary algorithm. An advantage of this approach is that the generated decision rules can be interpreted by users to extract further insight of the business processes. Furthermore, as previously mentioned, the method incorporates a new encoding based on event windows of different sizes which provides more information from event logs. Additionally, this method is accompanied by a full software stack we have developed to support both the training and the prediction phase of our predictive monitoring approach. The learning phase is supported by a ProM plugin that helps in the computation of process indicators and the preprocessing of the event log for the machine learning algorithm. The prediction phase is supported by a framework that enables the integration of run-time predictions obtained from the predictive models generated by the training phase with business process management systems like Camunda (Camunda, 2016).

Our approach was exhaustively tested with two different real-life event logs to assess the validity of the proposal. The datasets belong to IT Department of Health Services of Andalusia (Spain) and the BPI 2013 Challenge (Verbeek, 2013). For the validation of the proposal, we also include a comparison with a method of the literature, described in Breuker, Matzner, Delfmann, and Becker (2016), and several machine learning approaches, under the same experimental conditions, in order to justify the use of the evolutionary algorithm.

The remainder of this paper is organized as follows. Section 2 introduces the main concepts referred throughout the paper. Section 3 summarizes the related work in this area. Section 4 introduces our methodology. Section 5 presents the experimentation and obtained results. Finally, Section 6, includes some conclusions and possible future works.

2. Background on predictive monitoring

The goal of predictive monitoring is to predict some aspect of the execution of a running process instance. To do so, it relies on the existence of an event log that contains the relevant information of the execution of a business process. An event log (L) is composed of a set of traces (T) that contain each event (E) that occurs in the different instances of a business process. Each execution of a process instance is reflected in a trace. Formally, we can express a trace T_i as a list of events $T_i = [E_{i_1}, \dots, E_{i_m}]$ where E_{i_1} represents the first event and E_{i_m} reflects the final event of the execution of the process instance. An event (E) represents an instant of the execution of an activity of the process. Each event contains a set of attributes or properties (p) which represents all the information for the definition of such event, *e.g.* timestamp or the resource that execute a determined activity $E_j = [p_{j_1}, \dots, p_{j_n}]$ where n determines the total number of properties of the event. Finally, we can represent a log as a sequence of instances which have finished in an interval of time $L = [T_1, \dots, T_m]$ where T_1 represents the first executed trace and T_m is the last execution trace in the time interval.

Table 1
Event log example.

case id	event id	timestamp	activity	resource	cost
1	107561	12-12-2016:12.15	A	Lucas	100
	107562	12-12-2016:14.55	B	Lucas	300
	107563	12-12-2016:17.30	C	Paul	200
	107564	13-12-2016:12.15	D	Laura	400
2	108631	14-12-2016:10.00	A	Fred	100
	108632	14-12-2016:12.52	D	Fred	200
	108633	14-12-2016:13.27	E	Barney	100
3	108945	15-12-2016:10.32	B	Alan	100
	108946	16-12-2016:09.18	E	Sylvia	300

Table 1 depicts an example of event log. Each grouped row represents a trace. Therefore, the example shows 3 traces (case id: 1, 2 and 3) consisting of 4, 3 and 2 events, respectively (event id row). The number of events of each trace can be different. Finally, each event has several properties. In the table, four of them are depicted, namely: *timestamp, activity, resource* and *cost*.

A business process indicator (I) is a quantifiable metric that can be measured directly by data that is generated within the process flow. There are two types of process indicators: instance-level indicator and aggregated indicator (del Río-Ortega, Resinas, Cabanillas, & Ruiz-Cortés, 2013). An instance-level indicator provides a metric for a single process instance. It can be defined as a function of a trace T, *i.e.* $I(T)$, which is calculated by using the values of the attributes of the events that belong to this trace. This function can return a Boolean value, *e.g.* a determined condition fulfilled by the trace, or a real value, *e.g.* the duration of an activity. An aggregated indicator (I_A) can be represented as a function $I_A(L^f)$, where $L^f = \{T_i \in L \mid filter(T_i)\}$ contains all the traces T_i from an event log L that fulfill a determined requirement $filter(T_i)$. Generally, this filter is defined as a temporal constraint, such as an interval of time. In this work, we consider only instance-level indicators.

The goal of predictive monitoring is to predict the value of the indicator before a process instance finishes by means of a predictive model, which are usually built based on the information provided by the traces of previous process instances. Therefore, a predictive model for an indicator I is a function $P_I([E_{i_k}, \dots, E_{i_l}])$, with $k \le l$, that computes a prediction for I from the partial trace $[E_{i_k}, \dots, E_{i_l}]$, where E_{i_l} is the last event that have occurred in trace T_i at a given moment. If $k = 1$, then all events that have occurred in the process instance at hand are considered. Otherwise, if $k = l$, then only the last event of the process instance is considered.

There are two types of predictions that are usually relevant in the context of predictive monitoring, namely: next-event prediction and end-of-instance prediction. In next-event prediction, the goal is to predict the value of the indicator for the next event in the process instance ($E_{i_{l+1}}$), *i.e.*, $P_I([E_{i_k}, \dots, E_{i_l}])$ is approximately $I([E_{i_1}, \dots, E_{i_{l+1}}])$. On the other hand, in end-of-instance prediction, the goal is to predict the value of the indicator at the end of the process instance, *i.e.*, $P_I([E_{i_k}, \dots, E_{i_l}])$ is approximately $I([E_{i_1}, \dots, E_{i_m}])$, where E_{i_m} is the last event of the process instance.