The *FeaturePrediction* Package in ProM: Correlating Business Process Characteristics*

Massimiliano de Leoni and Wil M.P. van der Aalst

Eindhoven University of Technology, Eindhoven, The Netherlands
{m.d.leoni, w.m.p.v.d.aalst}@tue.nl

Abstract In Process Mining, often one is not only interested in learning process models but also in answering questions such as "What do the cases that are late have in common?", "What characterizes the workers that skip this check activity?" and "Do people work faster if they have more work?". Such questions can be answered by combining process mining with classification (e.g., decision tree analysis). Several authors have proposed ad-hoc solutions for specific questions, e.g., there is work on predicting the remaining processing time and recommending activities to minimize particular risks. This paper reports on a tool, implemented as plug-in for *ProM*, that unifies these ideas and provide a general framework for deriving and correlating process characteristics. To demonstrate the maturity of the tool, we show the steps with the tool to answer one correlation question related to a health-care process. The answer to a second question is shown in the screencast accompanying this paper.

1 Introduction

Process mining is not only about automatically learning process models. It also concerns with replaying event logs on the model to, e.g., check conformance or to uncover bottlenecks in the process. However, such analyses are often only the starting point for providing initial insights. When discovering a bottleneck or frequent deviation, one would like to understand why it exists. This requires the correlation of different *process characteristics*. These characteristics can be based on the control-flow (e.g., the next activity going to be performed), the data-flow (e.g., the amount of money involved), the time perspective (e.g., the activity duration or the remaining time to the end of the process), the organization perspective (e.g., the resource going to perform a particular activity), or, in case a normative process model exists, the conformance perspective (e.g., the skipping of a mandatory activity).

The study of these characteristics and how they influence each other is of crucial importance when an organization aims to improve and redesign its own processes. Many authors have proposed techniques to relate specific characteristics in an ad-hoc manner, such as to predict the remaining processing time of a case or to analyze routing decisions in the process or possible risks (see [1] for a detailed literature analysis). These problems are specific instances of a more general problem, which is concerned with *relating any process or event characteristic to other characteristics associated with single events or the entire process*. This paper reports on a tool that solves the more

* Dr. de Leoni conducted this work when also affiliated with University of Padua, Italy, and financially supported by the Eurostars - Eureka project PROMPT (E!6696).

general correlation problem. The tool unifies the ad-hoc approaches described in literature by providing a generic way to relate any characteristic (dependent variable) to other characteristics (independent variables). Readers are referred to [1] for a thorough introduction to the framework.

Starting point is an *event log*. For each process instance (i.e., case), there is a trace, i.e., a sequence of events. Events are associated with different *characteristics*, represented a key-value pairs. Mandatory characteristics are *activity* and *timestamp*. Other typical characteristics are the *resource* used to perform the activity, *transactional* information (start, complete, suspend, resume, etc.), and *costs*. However, many more characteristics can be associated to an activity (e.g., the age of a patient or size of an order).

The tool builds a table where each row corresponds to a different event and each column is a different characteristic. One of the columns become the dependent characteristic and the others are the independent characteristics; the relation between dependent and independent characteristics is discovered using decision-tree learning techniques. Before discovering the tree, the tool also allows some rows to be filtered out. For instance, one may want to only retain those events that refer to certain activities.

If a certain characteristic is valuable for an analysis but not present, our tool also allows extending event logs with additional characteristics that are not readily available. For instance, events can be extended with the remaining flow time till the end of the process instance or, also, the elapsed time since the process instance started. Other characteristics that may be added could be related to the resource who triggered an event (e.g., workload of the resource), i.e. who executed the respective activity. We can also add the next activity as a characteristic of an event. One can even add conformance checking results and external context information, such as weather information, to events as characteristics. In many cases, the values of these characteristics can be simply derived from the event log itself; in other cases, they need to be harvested from information sources outside the event log (weather information, stock index, etc.).

Implementation. The tool is implemented as a plug-in of ProM, an open-source "pluggable" framework for the implementation of process mining tools in a standardised environment (see http://www.promtools.org). The ProM framework is based on the concept of packages each of which is an aggregation of several plug-ins that are conceptually related. Our new plug-in is available in a new package named *FeaturePrediction*, which is available in ProM version 6.4.

A ProM plug-in requires a number of input objects and produces one or more output objects. The main input object of our plug-in is an event log, whereas the output is a decision tree. To build decision trees, the plug-in leverages on the implementation of the C4.5 algorithm in Weka (http://weka.sourceforge.net/). As mentioned before, our framework envisions the possibility to augment/manipulate the event logs with additional features. On the this concern, the tool is easily extensible: a new log manipulation can be easily plugged in by *(1)* implementing 3 methods in a Java class that inherits from an abstract class and *(2)* programmatically adding it to a given Java set of available log manipulations. To date, the implementation already includes an extensive number of manipulations, which cover different process perspectives (time, control-flow, data, resource and conformance) and are listed in Table 1 of [1]. The application of some log manipulations requires additional input objects, such as a process model or a LTL formula. The plug-in is organized in a way that one arbitrary additional object can be

given as input and used as source of information to enable log manipulations that can exploit it.

2 Usage of the Tool to Perform a Correlation Analysis Use Case

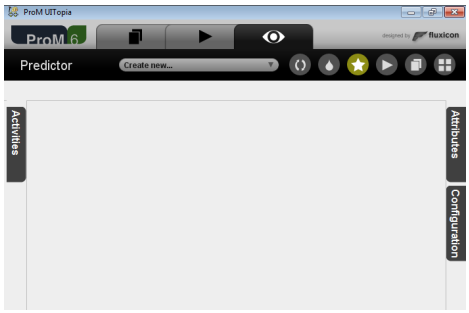


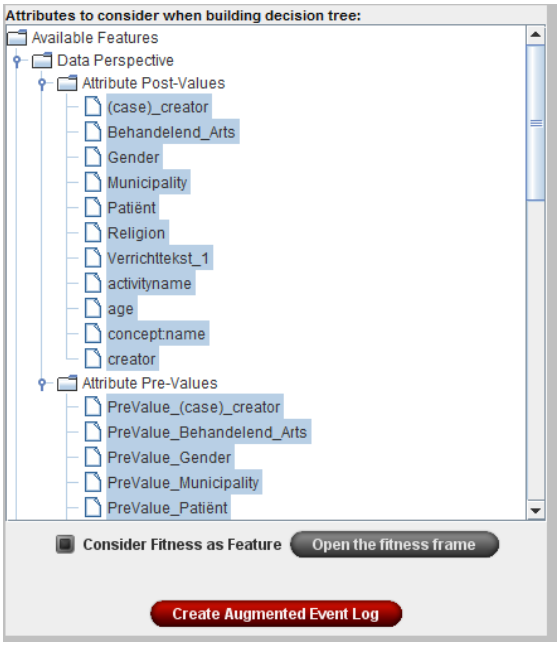
Figure 1. The starting screen of the tool.

In [1], we have reported on the application of our framework in collaboration of UWV, the Dutch institution that manages the provision of unemployment benefits for the employees in the Netherlands who had previously lost their job. In particular, we developed four analysis use cases to answer as many questions for which the institution was seeking an answer. As reported, many insights were derived, which had significant business value for UWV. However, in this paper, we want to complement such a evaluation with another one in a different business context. This section will show how an analysis use case can be carried out through our tool implementation in ProM. It is concerned with the process of treatment of pathologies related to eyes in a hospital in the Netherlands.

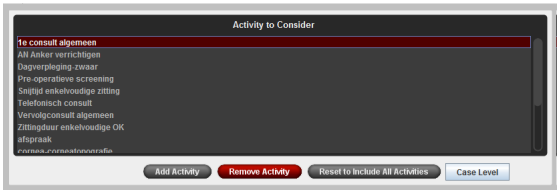
The analysis use case aims at correlating the duration of executing activity *Afspraak* (in Dutch, appointment) to other process characteristics. This activity is performed by physicians who periodically visit hospitalized patients. After starting ProM, the user needs to choose plug-in *Perform Prediction of Business Process Features*. In addition to giving an event log as input, we also put forward a second object that provides the necessary information to augment/manipulate events with characteristics linked to the conformance of process instances against a prescribed process model (see [2] for details). The initial screen is shown in Figure 1: no decision tree is constructed yet since the events to retain need to be chosen along with the dependent and independent characteristics to consider. The border of the screen contains three labels, namely *Activities*, *Attributes* and *Configuration*, used to, respectively, select activities for the events to retain, to pick the characteristics to consider and to set the parameters to construct the decision tree.

By passing over the labels with the mouse, different configuration panels are shown (see Figure 2) The first step concerns with choosing the characteristics to consider: Figure 2(a) shows the panel where users select the characteristics to consider among those available. These characteristics are visualized in a tree and grouped by the process perspective to which they refer. By selecting a node in a tree, characteristics are added to those to consider.

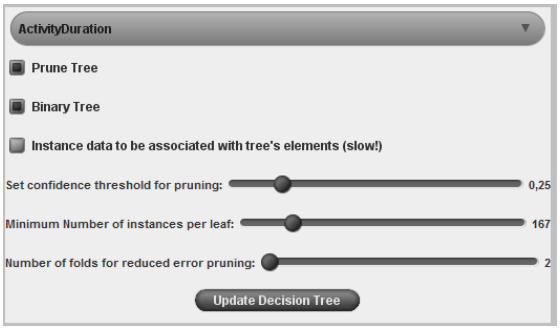
The characteristics linked to conformance are displayed differently: by selecting *Consider fitness as feature*, each event is augmented with the level of fitness of the trace to which the event belongs. By clicking on *Open the fitness frame*, users



(a) Panel to select the process characteristics to consider.



(b) Panel to filter on the activities of the events to retain.



(c) Panel to select the dependent characteristics and the parameters for the decision-tree construction.

Figure 2. Configuration Panels to build a correlation analysis use case.

can selectively decide (panel not shown here) if the number of deviations for certain single activities should be considered as characteristics (see [2] for more details). After choosing the characteristics to consider, the next step is about selecting the activities to retain. Since we aim to only provide correlation for *Afspraak*, events referring to any other activity are filtered out. Figure 2(b) shows the corresponding panel: any activity different from *Afspraak* is going to be removed from the list.

The filtering of events happens in the phase that follows the manipulation with additional characteristics. This means that the choice of events to retain does not influence how events are augmented with additional characteristics, e.g. referring to the number of executions of given activities or to the previous/next activity in trace. As final step, the analyst needs to choose which characteristic is the dependent one. This is done through the panel *Configuration*, shown in Figure 2(c). For our analysis use case, we selected *Activity Duration* as dependent characteristic.

The dependent characteristic needs to be one among those selected through the panel in Figure 2(a). The other options in the panel are used to configure the application of the C4.5 algorithm when building a decision tree. In particular, for this analysis, we decided to constrain the decision tree to be binary and allowed the decision tree to be

pruned, with the constraint that no less than 167 events can be associated with a leaf so as to balance under- and over-fitting problems. C4.5 requires a dependent characteristic to be discrete. The activity duration is a continuous characteristic and, hence, needs to be discretized before being used. Different discretization techniques are accessible through the *Discretization* panel (not shown here). For this analysis, we opted for *equal-frequency binning*: intervals are of different sizes but (roughly) the same number of observed values falls into each one.

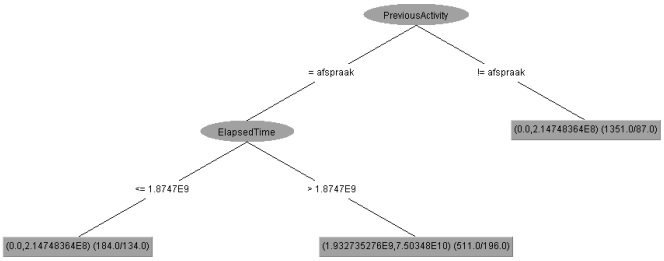


Figure 3. The resulting decision tree that provides a correlation with the duration of executions of activity *Afspraak*.

Figure 3 shows the resulting decision tree. Some correlation rules can be derived: if the previous activity is not *Afspraak*, the duration of an *Afspraak* execution is likely being less than 214,748,364 milliseconds, nearly 2.5 days. Similar durations are also expected for the executions of *Afspraak* preceded by another *Afspraak* when the patient treatments have started since less than 1,874,700,000 milliseconds, around 21.7 days. Conversely, the duration of *Afspraak* executions seems to be significantly longer, i.e. around 22.3 instead of 2.5 days, if the patient treatments have started since a longer time. Since the event log only stored the timestamp of completions of activities, this duration accounts for both the actual execution time and the waiting/idle time before *Afspraak* was actually started. If the event log also contained the timestamps when activities were started in cases, the duration would not consider the idle time. No correlation is made with characteristics related to resources and deviations. This means that the duration of the *Afspraak* executions is not related to those process characteristics.

At https://svn.win.tue.nl/repos/prom/Documentation/FeaturePrediction/screencast.avi, a screencast is available that, starting for the event log and the reference process model, shows the entire sequence of steps to obtain the decision tree in Figure 3. The screencast also reports on a different correlation analysis use case that is concerned with correlating several characteristics to the level of fitness of process instances with respect to given reference process model.

References

1. de Leoni, M., var der Aalst, W.M.P., Dees, M.: A General Framework for Correlating Business Process Characteristics. In: Proceedings of the 12th International Conference of Business Process Management (BPM 2014). Volume 8659 of LNCS., Springer (2014) 250–266
2. de Leoni, M., van der Aalst, W.M.P.: Aligning event logs and process models for multi-perspective conformance checking: An approach based on integer linear programming. In: Proceedings of the 11th International Conference on Business Process Management (BPM'13). Volume 8094 of LNCS., Springer-Verlag (2013) 113–129