Decision Mining in ProM

A. Rozinat and W.M.P. van der Aalst

Department of Technology Management, Eindhoven University of Technology
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands
{a.rozinat,w.m.p.v.d.aalst}@tm.tue.nl

Abstract. Process-aware Information Systems typically log events (e.g., in transaction logs or audit trails) related to the actual business process executions. Proper analysis of these execution logs can yield important knowledge that can help organizations to improve the quality of their services. Starting from a process model, which can be discovered by conventional process mining algorithms, we analyze how data attributes influence the choices made in the process based on past process executions. Decision mining, also referred to as decision point analysis, aims at the detection of data dependencies that affect the routing of a case. In this paper we describe how machine learning techniques can be leveraged for this purpose, and we present a *Decision Miner* implemented within the ProM framework.

1   Introduction

Process mining techniques have proven to be a valuable tool in order to gain insight into how business processes are handled within organizations. Taking a set of real process executions (the so-called "event logs") as the starting point, these techniques can be used for *process discovery* and *conformance checking*. Process discovery [2, 3] can be used to automatically construct a process model reflecting the behavior that has been observed and recorded in the event log. Conformance checking [1, 9] can be used to compare the recorded behavior with some already existing process model to detect possible deviations. Both may serve as *input* for designing and improving business processes, e.g., conformance checking can be used to find problems in existing processes, and process discovery can be used as a starting point for process analysis and system configuration. While there are several process mining algorithms that deal with the control flow perspective of a business process [2] *less attention has been paid to how the value of a data attribute may affect the routing of a case.*

Most information systems (cf. WFM, ERP, CRM, SCM, and B2B systems) provide some kind of *event log* (also referred to as transaction log or audit trail) [2] where an event refers to a case (i.e., process instance) and an activity, and, in most systems, also a timestamp, a performer, and some additional data. Nevertheless, many process mining techniques only make use of the first two

attributes in order to construct a process model which reflects the causal relations that have been observed among the activities. In addition, machine learning algorithms have become a widely adopted means to extract knowledge from vast amounts of data [7, 11]. In this paper we use the well-known concept of *decision trees* to carry out a *decision point analysis*, i.e., to find out which properties of a case might lead to taking certain paths in the process. Starting from a discovered process model (i.e., a model discovered by conventional process mining algorithms), we try to enhance the model by integrating patterns that can be observed from data modifications, i.e., every choice in the model is analyzed and, if possible, linked to properties of individual cases and activities.

Clearly, the application of (existing) data mining techniques in the context of business processes has the potential to gain knowledge, or to make tacit knowledge explicit. Besides data attributes, resource information, and timestamps, even more general quantitative (e.g., key performance indicators like waiting time derived from the log) and qualitative (i.e., desirable or undesirable properties) information could be included in the analysis if available. To directly support data analysis for business processes we have implemented a *Decision Miner* in the context of the ProM framework[1], which offers a wide range of tools related to process mining and process analysis.

The paper is organized as follows. First, the use of machine learning techniques in the context of the decision point analysis is described in Section 2. Section 3 presents the Decision Miner plug-in of the ProM framework. Finally, related work is discussed in Section 4, and the paper concludes by pointing out future research directions.

2   Using Decision Trees for Analyzing Choices

In order to analyze the choices in a business process we first need to identify those parts of the model where the process is split into alternative branches, also called decision points. Based on data attributes associated to the cases in the event log we subsequently want to find rules for following one route or the other.

In terms of a Petri net, a decision point corresponds to a place with multiple outgoing arcs. Since a token can only be consumed by one of the transitions connected to these arcs, alternative paths may be taken during the execution of a process instance. In order to analyze the choices that were made in past process executions we need to find out which alternative branch was taken by a certain process instance. Therefore, the set of possible decisions must be described with respect to the event log. Starting from the identification of a choice construct in the process model a decision can be detected if the execution of an activity in the respective alternative branch of the model has been observed, which requires a mapping from that activity to its "occurrence footprint" in the event log. So, if a process instance contains the given "footprint", this means that there was a decision for the associated alternative path in the process. For

---

[1] Both documentation and software (including the source code) can be downloaded from *www.processmining.org*.

simplicity we examine the occurrence of the *first* activity per alternative branch in order to classify the possible decisions. However, in order to make decision mining operational for real-life business processes several challenges posed by, for example, *invisible activities*, *duplicate activities*, and *loops* need to be met. Because of the limited space here we refer the interested reader to our technical report [10], where these issues are addressed.

After identifying a decision point in a business process, the next step is to determine whether this decision might be influenced by case data, i.e., whether cases with certain properties typically follow a specific route. The idea is to convert every decision point into a *classification problem* [7, 11, 8], where the *classes* are the different decisions that can be made. As training examples we use the process instances in the log (for which it is already known which alternative path they followed with respect to the decision point). The attributes to be analyzed are the case attributes contained in the log, and we assume that all attributes that have been written *before* the considered choice construct are relevant for the routing of a case at that point. In order to solve such a classification problem there are various algorithms available [7, 11]. We decided to use decision trees (such as C4.5 [8]), which are among the most popular inductive inference algorithms, and which provide a number of extensions that are important for practical applicability. For example, they are able to deal with continuous-valued attributes, missing attribute values, and they include effective methods to avoid *overfitting* the data (i.e., that the tree is over-tailored towards the training examples). In [10] we show in detail how the training examples and the inferred decision rules look for each decision point in the example process.

3   Decision Mining with the ProM Framework

The approach presented in this paper was implemented as a plug-in for the ProM Framework. The *Decision Miner* plug-in determines the decision points contained in a Petri net model[2], and specifies the possible decisions with respect to the log while being able to deal with invisible and duplicate activities in the way described in [10]. Figure 1(a) shows the Model view of the Decision Miner, which provides a visualization of each decision point with respect to the given process model. The example process sketches the processing of a liability claim within an insurance company, and the depicted decision point relates to the choice of whether a full check or a policy-only check is to be performed. Only attributes of the activity "Register Claim" are within the analysis scope of this decision point. The Attributes view shown in Figure 1(b) allows for the selection of those attributes to be included in the analysis of each decision point. During the execution of activity "Register claim" information about the amount of money involved (*Amount*), the corresponding customer (*CustomerID*), and the type of policy (*PolicyType*) are provided. We retain all these attributes

---

[2] Note that although only Petri net process models are directly supported by the Decision Miner, various other process model types (EPC, YAWL, etc.) are indirectly supported via conversion tools available in ProM.
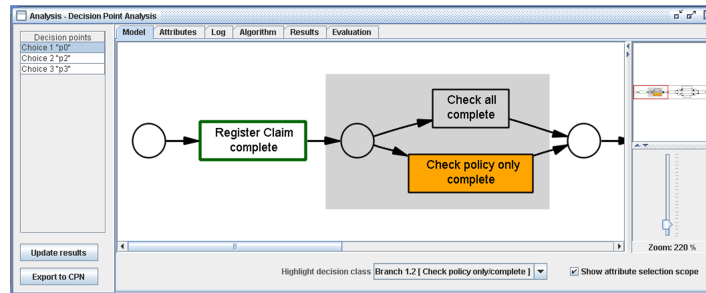
(a) Model view visualizes decision points in the process model

(b) Attributes view allows for selection of those attributes to be used for analysis

Fig. 1. Screenshots of the the Decision Miner in ProM

and set the type of the *Amount* attribute as numeric. Here the advantage of a tool suite like ProM becomes visible. The tight integration of further analysis components available in the framework can be used to add meta data to the event log before starting the actual decision point analysis. For example, a previous performance analysis evaluating the timestamps of each log event can provide additional attributes, such as the flow time and waiting time, to specific activities or to the whole process instance. These attributes then become available for analysis in the same way as the initial attributes.

While the Decision Miner formulates the learning problem, the actual analysis is carried out with the help of the J48 decision tree classifier, which is the implementation of the C4.5 algorithm [8] provided by the Weka software library [11]. The Algorithm view offers the full range of parameters that are available for the used decision tree algorithm from the Weka library. Figure 2(a) shows the decision tree result for the decision point *p0*, from which we can now infer the logical expressions that form the decision rules depicted in Figure 2(b) in the following way: If an instance is located in one of the leaf nodes of a decision tree, it fulfills all the predicates on the way from the root to the leaf, i.e., they are connected by a boolean AND operator; When a decision class is represented by multiple leaf nodes in the decision tree the leaf expressions are combined via a boolean OR operator. The discovered rules indicate that the extensive check (activity *B*) is only performed if the *Amount* is greater than 500 and the *Pol-*
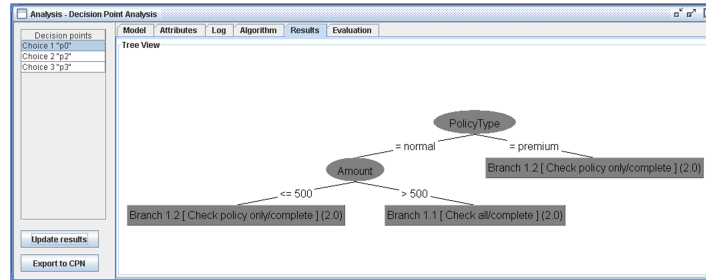
(a) Decision tree result for analysis of decision point "p0"
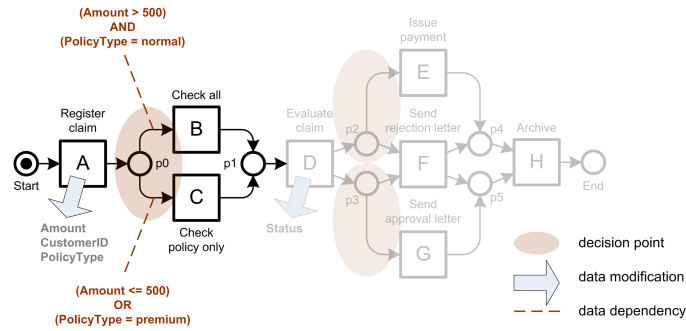
(b) Enhancing the process model by the discovered rules

Fig. 2. Interpreting the decision tree result

icyType is "normal", whereas a simpler coverage check (*C*) is sufficient if the *Amount* is smaller than or equal to 500, or the *PolicyType* is "premium" (which may be due to certain guarantees from "premium" member corporations).

4 Related Work

The work reported in this paper is closely related to [5], in which the authors describe the architecture of the *Business Process Intelligence* (BPI) tool suite situated on top of the *HP Process Manager* (HPPM). Whereas they outline the use of data mining techniques for process behavior analysis in a broader scope, we show how a decision point analysis can be carried out in conjunction with process mining, i.e., we do not assume some a priori model. Another important difference, although not presented in this paper, is that we can also analyze models in the presence of duplicate and invisible activities. In [6] decision trees are used to analyze staff assignment rules. Additional information about the organizational structure is incorporated in order to derive higher-level attributes (i.e., roles) from the actual execution data (i.e., performers). In [4] the authors aim at the integration of neural networks into EPC process models via fuzzy events and fuzzy functions. While this approach may support, e.g., one concrete mortgage grant decision process, we focus on the use of machine learning techniques as a general tool to analyze business process executions.

5 Conclusion

In this paper we have presented a *Decision Miner* that analyzes the choice constructs of a (mined) Petri net process model in the context of the ProM framework. Future research plans include the support of further types of process models (such as EPCs), and the provision of alternative algorithms already available in the data mining field (and related software libraries). For example, sometimes it is better to directly capture a concept description in rules rather than in a decision tree. Finally, the application of data mining techniques in the context of business processes can be beneficial beyond the analysis of decisions that have been made. A free specification of the learning problem on the available data can be used to, for example, mine association rules, or to assess potential correlations to the fact that a case has a throughput time which exceeds some user-defined threshold value.

References

1. W.M.P. van der Aalst. Business Alignment: Using Process Mining as a Tool for Delta Analysis. In J. Grundspenkis and M. Kirikova, editors, *Proceedings of the 5th Workshop on Business Process Modeling, Development and Support (BPMDS'04)*, volume 2 of *Caise'04 Workshops*, pages 138–145. Riga Technical University, Latvia, 2004.
2. W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data and Knowledge Engineering*, 47(2):237–267, 2003.
3. W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.
4. O. Adam, O. Thomas, and P. Loos. Soft Business Process Intelligence — Verbesserung von Geschäftsprozessen mit Neuro-Fuzzy-Methoden. In F. Lehner et al., editor, *Multikonferenz Wirtschaftsinformatik 2006*, pages 57–69. GITO-Verlag, Berlin, 2006.
5. D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M.-C. Shan. Business Process Intelligence. *Computers in Industry*, 53(3):321–343, 2004.
6. L. T. Ly, S. Rinderle, P. Dadam, and M. Reichert. Mining Staff Assignment Rules from Event-Based Data. In C. Bussler et al., editor, *Business Process Management 2005 Workshops*, volume 3812 of *Lecture Notes in Computer Science*, pages 177–190. Springer-Verlag, Berlin, 2006.
7. T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
8. J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
9. A. Rozinat and W.M.P. van der Aalst. Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models. In C. Bussler et al., editor, *Business Process Management 2005 Workshops*, volume 3812 of *Lecture Notes in Computer Science*, pages 163–176. Springer-Verlag, Berlin, 2006.
10. A. Rozinat and W.M.P. van der Aalst. Decision Mining in Business Processes. BPM Center Report BPM-06-10, BPMcenter.org, 2006.
11. I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann, 2005.