The Need for a Process Mining Evaluation Framework in Research and Practice

Position Paper

A. Rozinat, A.K. Alves de Medeiros, C.W. Günther,
A.J.M.M. Weijters, and W.M.P. van der Aalst

Eindhoven University of Technology
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands
{a.rozinat,a.k.medeiros,c.w.gunther,a.j.m.m.weijters,
w.m.p.v.d.aalst}@tue.nl

Abstract. Although there has been much progress in developing process mining algorithms in recent years, no effort has been put in developing a common means of assessing the quality of the models discovered by these algorithms. In this paper, we motivate the need for such an evaluation mechanism, and outline elements of an evaluation framework that is intended to enable (a) process mining researchers to compare the performance of their algorithms, and (b) end users to evaluate the validity of their process mining results.

1 Introduction

Process mining has proven to be a valuable approach that provides new and objective insights into the way business processes are actually conducted within organizations. Taking a set of real executions (the so-called "event log") as the starting point, these techniques attempt to extract non-trivial and useful process information from various perspectives, such as control flow, data flow, organizational structures, and performance characteristics. A common *mining XML* (MXML) log format was defined in [3] to enable researchers and practitioners to share their logs in a standardized way. However, while process mining has reached a certain level of maturity and has been used in a variety of real-life case studies (see [1] for an example), *a common framework to evaluate process mining results is still lacking*. We believe that there is the need for a concrete framework that enables (a) process mining researchers to compare the performance of their algorithms, and (b) end users to evaluate the validity of their process mining results. This paper is a first step into this direction.

The driving element in the process mining domain is some operational process, for example a business process such as an insurance claim handling procedure in an insurance company, or the booking process of a travel agency. Nowadays, many business processes are supported by information systems that help coordinating the steps that need to be performed in the course of the process. Workflow systems, for example, assign work items to employees according to their roles

A. ter Hofstede, B. Benatallah, and H.-Y. Paik (Eds.): BPM 2007 Workshops, LNCS 4928, pp. 84–89, 2008.

and the status of the process. Typically, these systems record *events* related to the activities that are performed, e.g., in audit trails or transaction logs [3].[1] These event logs form the input for process mining algorithms.

In this paper we focus on providing a means of comparison for algorithms that discover the *control-flow perspective* of a process (which we simply refer to as process discovery algorithms from now on). In particular, we focus on *validation techniques* for these process discovery algorithms. We argue that this evaluation can take place in different dimensions, and identify ingredients that are needed for an evaluation framework. Note that in an extended version of this paper [11] we describe two different validation approaches: one based on existing validation metrics, and another based on the so-called k-fold cross validation technique known from the machine learning domain. We applied both approaches to the running example. Furthermore, in [11] we also present an extensible `Control Flow Benchmark` plug-in to directly support the evaluation and comparison of different mining results in the context of the ProM framework[2].

The remainder of this paper is organized as follows. Section 2 motivates the need for an evaluation framework. Then, Section 3 outlines first steps towards such a common framework. Finally, Section 4 concludes the paper.

2 Process Discovery: Which Model Is the "Best"?

The goal of a process discovery algorithm is to construct a process model which reflects the behavior that has been observed in the event log. Different process modeling languages[3] can be used to capture the causal relationships of the steps, or activities, in the process. The idea of applying process mining in the context of workflow management was first introduced in [5]. Over the last decade many process mining approaches have been proposed [6,9]. While all these approaches aim at the discovery of a "good" process model, often targeting particular challenges (e.g., the mining of loops, or duplicate tasks), they have their limitations and many different event logs and quality measurements are used. Hence, no standard measure is available.

To illustrate the dilemma, we consider the simple example log in Figure 2(a), which contains only five different traces. We applied six different process mining algorithms that are available in ProM and obtained six different process models (for every plug-in, we used the default settings in ProM 4.1). Figure 1 depicts the mining results for the `Alpha` miner [4], the `Heuristic` miner [12], the `Alpha++`

[1] It is important to note that information systems that do not enforce users to follow a particular process often still provide detailed event logs, e.g., hospital information systems, ERP systems etc.

[2] ProM offers a wide range of tools related to process mining and process analysis. Both documentation and software (including the source code) can be downloaded from *http://www.processmining.org*.

[3] In the remainder of this paper we will use Petri nets, motivated by their formal semantics. Note that in our tool ProM there exist translations from process modeling languages such as EPC, YAWL, and BPEL to Petri nets and vice-versa.

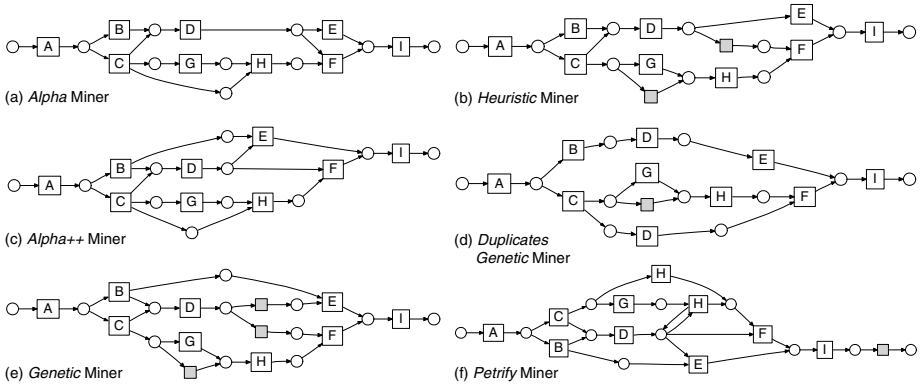

Fig. 1. Process models that were discovered by different process discovery algorithms based on the same log

miner [13], the `Duplicates Genetic` miner and the `Genetics` miner [8], and the `Petrify` miner [2]. The models seem similar, but are all different[4]. Are they equivalent? If not, which one is the "best"?

These questions are interesting both for researchers and end users: (a) Researchers typically attempt to let their process discovery algorithms construct process models that completely and precisely reflect the observed behavior in a structurally suitable way. It would be useful to have common data sets containing logs with different characteristics, which can be used within the scientific community to systematically compare the performance of various algorithms in different, controlled environments. (b) Users of process discovery techniques, on the other hand, need to know how well the discovered model describes reality, how many cases are actually covered by the generated process description etc. For example, if in an organization process mining is to be used as a knowledge discovery tool in the context of a Business Process Intelligence (BPI) framework, it must be possible to estimate the "accuracy" of a discovered model, i.e., the "confidence" with which it reflects the underlying process. Furthermore, end users need to be able to compare the results obtained from different process discovery algorithms.

3 Towards a Common Evaluation Framework

In an experimental setting, we usually know the original model that was used to generate an event log. For example, the log in Figure 2(a) was created from the simulation of the process model depicted in Figure 2(b). Knowing this, one could leverage *process equivalence* notions to evaluate the discovered model with respect

[4] Note that throughout this paper the invisible (i.e., unlabeled) tasks need to be interpreted using the so-called "lazy semantics", i.e., they are only fired if they enable a succeeding, visible task [8].

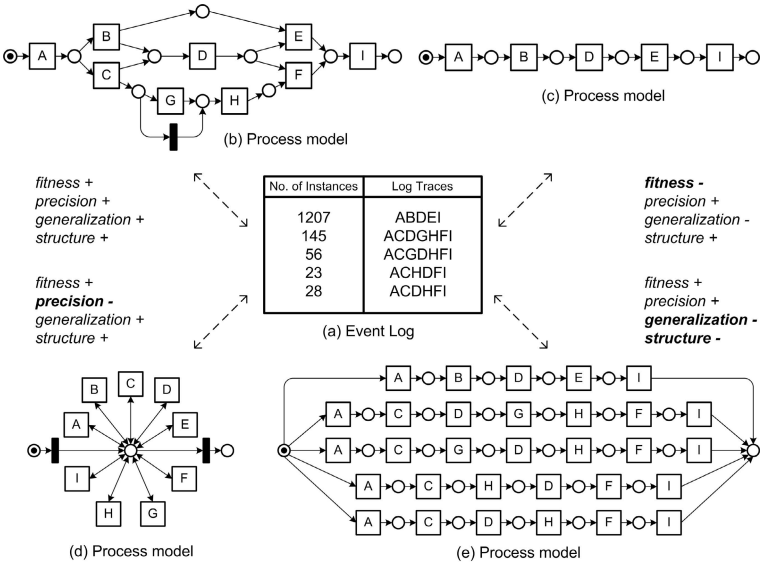

Fig. 2. The evaluation of a process model can take place in different dimensions

to the original model. But in many practical situations no original model is available. However, if we assume that the behavior observed in the log is what really happened (and somehow representative for the operational process at hand), it is possible to compare the discovered model to the event log that was used as input for the discovery algorithm. This essentially results in a *conformance analysis* problem [10,7]. In either case quality criteria need to be determined.

Evaluation Dimensions. Figure 2 depicts an event log (a) and four different process models (b-e). While Figure 2(b) depicts a "good" model for the event log in Figure 2(a), the remaining three models show undesirable, extreme models that might also be returned by a process mining algorithm. They illustrate that the evaluation of an event log and a process model can take place in different, orthogonal dimensions.

Fitness. The first dimension is fitness, which indicates how much of the observed behavior is captured by (i.e., "fits") the process model. For example, the model in Figure 2(c) is only able to reproduce the sequence *ABDEI*, but not the other sequences in the log. Therefore, its fitness is poor.

Precision. The second dimension addresses overly general models. For example, the model in Figure 2(d) allows for the execution of activities $A - I$ in any order (i.e., also the sequences in the log). Therefore, the fitness is good, but the precision is poor. Note that the model in Figure 2(b) is also considered to be a precise model, although it additionally allows for the trace *ACGHDFI* (which is not in the log). Because the number of possible sequences generated by a process model may grow exponentially, it is not likely that all the possible behavior has been observed

in a log. Therefore, process mining techniques strive for weakening the notion of *completeness* (i.e., the amount of information a log needs to contain to be able to rediscover the underlying process [4]). For example, they want to detect parallel tasks without the need to observe every possible interleaving between them.

Generalization. The third dimension addresses overly precise models. For example, the model in Figure 2(e) only allows for *exactly* the five sequences from the log. In contrast to the model in Figure 2(b) no generalization was performed. Determining the right level of generalization remains a challenge, especially when dealing with logs that contain *noise* (i.e., distorted data). Similarly, in the context of more unstructured and/or flexible processes, it is essential to further abstract from less important behavior (i.e., restriction rather than generalization). In general, abstraction can lead to the omission of connections between activities, which could mean lower precision or lower fitness (e.g., only capturing the most frequent paths). Furthermore, steps in the process could be left out completely. Therefore, abstraction must be seen as a different evaluation dimension, which needs to be balanced against precision and fitness.

Structure. The last dimension is the structure of a process model, which is determined by the vocabulary of the modeling language (e.g., routing nodes with AND and XOR semantics). Often there are several syntactic ways to express the same behavior, and there may be "preferred" and "less suitable" representations. For example, the fitness and precision of the model in Figure 2(e) are good, but it contains many duplicate tasks, which makes it difficult to read. Clearly, this evaluation dimension highly depends on the process modeling formalism, and is difficult to assess in an objective way as it relates to human modeling capabilities.

Evaluation Framework. To systematically compare process mining algorithms, it would be useful to have common data sets, which can be used and extended by different researchers to "benchmark" their algorithms on a per-dataset basis. For instance, in the machine learning community there are well know data sets (e.g., the UCI Machine Learning Repository, CMU NN-Bench Collection, Proben1, Stat-Log, ELENA-data, etc.) that can be used for testing and comparing different techniques. Such a process mining repository could be seen as an element in a possible evaluation framework, and should also provide information about the process or log characteristics as these may pose special challenges. Furthermore, the results of an evaluation could be stored for later reference.

At the same time it is necessary to be able to influence both the process and log characteristics. For example, one might want to generate an event log containing noise (i.e., distorting the logged information), or a certain timing behavior (some activities taking more time than others), from a given model. For log generation, simulation tools such as CPN Tools can be used. Another example for log generation is the generation of "forbidden" scenarios as a complement to the actual execution log.

Clearly, many different approaches for evaluation and comparison of the discovered process models are possible. As a first step, in [11] we have looked at existing evaluation techniques both in the process mining and data mining domain.

4 Conclusion

Adequate validation techniques in the process mining domain are needed to evaluate and compare discovered process models both in research and practice. Many obstacles such as bridging the gap between different modeling languages, defining good validation criteria and metrics for the quality of a process model etc. remain, and should be subject to further research. Moreover, a comprehensive set of benchmark examples is needed.

References

1. van der Aalst, W.M.P., Reijers, H.A., Weijters, A.J.M.M., van Dongen, B.F., Alves de Medeiros, A.K., Song, M., Verbeek, H.M.W.: Business Process Mining: An Industrial Application. Information Systems 32(5), 713–732 (2007)
2. van der Aalst, W.M.P., Rubin, V., van Dongen, B.F., Kindler, E., Günther, C.W.: Process Mining: A Two-Step Approach using Transition Systems and Regions. BPM Center Report BPM-06-30, BPMcenter.org (2006)
3. van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: Workflow Mining: A Survey of Issues and Approaches. Data and Knowledge Engineering 47(2), 237–267 (2003)
4. van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering 16(9), 1128–1142 (2004)
5. Agrawal, R., Gunopulos, D., Leymann, F.: Mining Process Models from Workflow Logs. In: Sixth International Conference on Extending Database Technology, pp. 469–483 (1998)
6. Cook, J.E., Wolf, A.L.: Discovering Models of Software Processes from Event-Based Data. ACM Transactions on Software Engineering and Methodology 7(3), 215–249 (1998)
7. Cook, J.E., Wolf, A.L.: Software Process Validation: Quantitatively Measuring the Correspondence of a Process to a Model. ACM Transactions on Software Engineering and Methodology 8(2), 147–176 (1999)
8. Alves de Medeiros, A.K.: Genetic Process Mining. PhD thesis, Eindhoven University of Technology, Eindhoven (2006)
9. Herbst, J.: A Machine Learning Approach to Workflow Management. In: Lopez de Mantaras, R., Plaza, E. (eds.) ECML 2000. LNCS (LNAI), vol. 1810, pp. 183–194. Springer, Berlin (2000)
10. Rozinat, A., van der Aalst, W.M.P.: Conformance Checking of Processes Based on Monitoring Real Behavior. Information Systems 33(1), 64–95 (2008)
11. Rozinat, A., Alves de Medeiros, A.K., Günther, C.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: Towards an Evaluation Framework for Process Mining Algorithms. BPM Center Report BPM-07-06, BPMcenter.org (2007)
12. Weijters, A.J.M.M., van der Aalst, W.M.P.: Rediscovering Workflow Models from Event-Based Data using Little Thumb. Integrated Computer-Aided Engineering 10(2), 151–162 (2003)
13. Wen, L., Wang, J., Sun, J.G.: Detecting Implicit Dependencies Between Tasks from Event Logs. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) APWeb 2006. LNCS, vol. 3841, pp. 591–603. Springer, Heidelberg (2006)