Process Mining: On the Balance Between Underfitting and Overfitting

W.M.P. van der Aalst

Eindhoven University of Technology,
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands.
w.m.p.v.d.aalst@tue.nl

Abstract. Process mining techniques attempt to extract non-trivial and useful information from event logs. One aspect of process mining is *control-flow discovery*, i.e., automatically constructing a process model (e.g., a Petri net) describing the causal dependencies between activities. One of the essential problems in process mining is that *one cannot assume to have seen all possible behavior*. At best, one has seen a representative subset. Therefore, classical synthesis techniques are not suitable as they aim at finding a model that is able to *exactly reproduce the log*. Existing process mining techniques try to avoid such "overfitting" by generalizing the model to allow for more behavior. This generalization is often driven by the representation language and very crude assumptions about completeness. As a result, parts of the model are "overfitting" (allow only what has actually been observed) while other parts may be "underfitting" (allow for much more behavior without strong support for it). This talk will present the main challenges posed by real-life applications of process mining and show that it is possible to balance between overfitting and underfitting in a controlled manner.

1 Process Mining

More and more information about processes is recorded by information systems in the form of so-called "event logs". Despite the omnipresence and richness of these event logs, most software vendors use this information for answering only relatively simple questions *under the assumption that the process is fixed and known*, e.g., the calculation of simple performance metrics like utilization and flow time. However, in many domains processes are evolving and people typically have an oversimplified and incorrect view of the actual business processes. Hence, the goal of *process mining* [1] is to learn about processes by observing them through event logs.

Process mining addresses the problem that most organizations have very limited information about what is actually happening in their organization. In practice, there is often a significant gap between what is prescribed or supposed to happen, and what *actually* happens. Three basic types of process mining can be identified: (1) *discovery*, (2) *conformance* checking, and (3) model *extension*.

These three types may be applied to the different perspectives of business processes, e.g., the *control-flow perspective*, the *case/data perspective*, the *resource perspective*, etc.

Although it is important to view process mining in a broader context, most interesting and most challenging is the discovery of the control-flow perspective. Today, there are many techniques that, based on an event log, are able to construct a process model. For example, using the α-algorithm [3] a Petri net can be discovered based on sequences of events. A tool like ProM offers a wide variety of control-flow discovery algorithms (cf. www.processmining.org).

2 Balancing Between Underfitting and Overfitting

ProM has been applied in several hospitals (AMC and Catherina hospitals), banks (ING), high-tech system manufacturers (ASML and Philips Medical Systems), software repositories for open-source projects, several municipalities (Heusden, Alkmaar, etc.), etc. These experiences show that *the main problem is finding a balance between "overfitting" and "underfitting"*. Some algorithms have a tendency to over-generalize, i.e., the discovered model allows for much more behavior than actually recorded in the log. The reason for over-generalizing is often the representation used and a coarse completeness notion. Other algorithms have a tendency to "overfit" the model. Classical synthesis approaches such as the "theory of regions" aim at a model that is able to exactly reproduce the log. Therefore, the model is merely another representation of the log without deriving any new knowledge.

This talks will focus on finding a balance between "overfitting" and "underfitting". One approach is to do process mining in two steps [2]. In the first step, a transition system is constructed. While constructing the transition system one can choose from various abstractions. In the second step, the transition system is transformed into a process model. This step is needed because the transition system is not able to show concurrency and parallel branches typically result in an explosion of states making the transition system unreadable. Hence, the goal of the second step is to provide a compact representation of the selected behavior. Note that the first step is mainly concerned with abstraction, while the second step is mainly concerned with representation issues.

References

1. W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, and H.M.W. Verbeek. Business Process Mining: An Industrial Application. *Information Systems*, 32(5):713–732, 2007.
2. W.M.P. van der Aalst, V. Rubin, B.F. van Dongen, E. Kindler, and C.W. Günther. Process Mining: A Two-Step Approach using Transition Systems and Regions. BPM Center Report BPM-06-30, BPMcenter.org, 2006.
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.