Information Systems 000 (2018) 1–16

Contents lists available at ScienceDirect

Information Systems

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

To aggregate or to eliminate? Optimal model simplification for improved process performance prediction

Arik Senderovich[a,*], Alexander Shleyfman[a], Matthias Weidlich[b,*], Avigdor Gal[a], Avishai Mandelbaum[a]

[a] Technion - Israel Institute of Technology, Haifa, Israel
[b] Humboldt-Universität zu Berlin, Berlin, Germany

ARTICLE INFO

Article history:
Received 19 December 2016
Revised 7 March 2018
Accepted 12 April 2018
Available online xxx

Keywords:
Generalised stochastic Petri nets
Model Simplification
Folding
Elimination
Aggregation
Process Mining

ABSTRACT

Operational process models such as generalised stochastic Petri nets (GSPNs) are useful when answering performance questions about business processes (e.g. 'how long will it take for a case to finish?'). Recently, methods for process mining have been developed to discover and enrich operational models based on a log of recorded executions of processes, which enables evidence-based process analysis. To avoid a bias due to infrequent execution paths, discovery algorithms strive for a balance between over-fitting and under-fitting regarding the originating log. However, state-of-the-art discovery algorithms address this balance solely for the control-flow dimension, neglecting the impact of their design choices in terms of performance measures. In this work, we thus offer a technique for controlled performance-driven model reduction of GSPNs, using structural simplification rules, namely *foldings*. We propose a set of foldings that aggregate or eliminate performance information. We further prove the soundness of these foldings in terms of stability preservation and provide bounds on the error that they introduce with respect to the original model. Furthermore, we show how to find an optimal sequence of simplification rules, such that their application yields a minimal model under a given error budget for performance estimation. We evaluate the approach with two real-world datasets from the healthcare and telecommunication domains, showing that model simplification indeed enables a controlled reduction of model size, while preserving performance metrics with respect to the original model. Moreover, we show that aggregation dominates elimination when abstracting performance models by preventing under-fitting due to information loss.

1. Introduction

Performance analysis is an important pillar of business process management initiatives in diverse domains, from telecommunication, through healthcare, to finance. Taking healthcare as an example, it involves the ability to answer questions such as 'how long will it take for a patient to get treatment?', and 'how many nurses do we need to staff to accommodate the incoming demand?' [1]. In call centers, analyzing performance drives the number of staffed agents, which correlates with vast operational costs [2]. Hence, answers to performance questions are key in running an organisation successfully and deliver value to its customers [3].

Operational process models such as generalised stochastic Petri nets and queueing networks are useful to answer the aforementioned performance questions [4,5]. In particular, these models enable testing of re-design and improvement initiatives with respect to the as-is model. For instance, by changing staffing levels and altering the control-flow, the impact of operational changes on the performance characteristics of the process can be explored.

Process mining enables automatic discovery and enrichment of operational process models from logs, which record process executions [6]. Data-driven model discovery improves beyond the manual model elicitation in its ability to accurately reflect the executed process. However, automatically discovered models tend to incorporate infrequent process executions, which may result in over-fitting with respect to the originating log. Recently proposed discovery algorithms attempt to balance between over-fitting and under-fitting in the control-flow dimension [7–10]. Yet, the question of how to balance over-fitting and under-fitting in terms of performance annotations of operational models has received little attention in the literature so far [11].

This work tackles the problem of balancing over- and under-fitting in performance-oriented models. Our method involves

* Corresponding authors.
E-mail addresses: sariks@tx.technion.ac.il (A. Senderovich), alesh@ie.technion.ac.il (A. Shleyfman), matthias.weidlich@hu-berlin.de (M. Weidlich), avigal@ie.technion.ac.il (A. Gal), avim@ie.technion.ac.il (A. Mandelbaum).

https://doi.org/10.1016/j.is.2018.04.003
0306-4379/

the automated simplification of generalised stochastic Petri nets (GSPNs) by starting with an initial GSPN discovered from a log, and applying simplification rules that generalise the model. To avoid under-fitting, we quantify the expected error incurred by every simplification, thereby linking model size and the total error in estimating the performance characteristics of a process.

The paper builds upon our earlier work [11] and provides three main contributions:

(1) *Structural foldings*: We define a set of structural simplification rules for GSPNs, referred to as *foldings*. Unlike existing proposals for model simplification [12], these rules are local (affecting only a subnet of the GSPN), come with formal bounds regarding the introduced estimation error, and their applicability is identified automatically by structural decomposition of the GSPN. Foldings, as a form of simplification, may either use aggregation or model elimination.
(2) *Theoretical foundation*: For each folding, we prove that it is proper, i.e. preserving queueing stability, a key property of performance models. Further, for each folding, we provide an error bound on the bias that the folding introduces with respect to the original model.
(3) *Optimality*: We formulate model simplification as an optimisation problem that aims at attaining a minimal model for a given cost budget for the introduced estimation error. We prove that the optimisation problem boils down to the well-established *tree-knapsack* problem, which can be cast as an Integer Linear Programming (ILP) problem. This enables efficient computation of the optimal sequence of folding operations.

We evaluate our approach with use-cases from two relevant domains: healthcare and telecommunication services. Our experiments show that simplification of a GSPN discovered from a real-world log enables users to balance over-fitting and under-fitting from the performance perspective.

The remainder of the paper is structured as follows. The next section discusses the methods and challenges in performance-oriented process mining. Section 3 recalls the GSPN formalism. Foldings of GSPNs are introduced in Section 4, while Section 5 discusses how to identify applicable foldings for a GSPN. The model simplification problem and its encoding as an ILP program is given in Section 6. Evaluation results are presented in Section 7. Section 8 reviews related work, before Section 9 concludes the paper.

2. Background: performance-oriented process mining

In this section, we provide an overview of techniques for performance-oriented process mining. We start with a brief review of the use of process mining for operational analysis (Section 2.1). Then, we provide the intuition for our model simplification technique as a method for alleviating overfitting in performance analysis of business processes (Section 2.2).

2.1. Process mining for operational analysis

We consider a setting in which a log L of recorded process executions is given and analysis questions regarding the performance of process execution shall be answered. Specifically, let Y be a performance measure, e.g., the total runtime of a process instance. Further, let $q(Y)$ be a performance query over Y, e.g., the expected value of Y, which we aim at answering based on L. In general, we distinguish two types of process mining techniques to quantify $q(Y)$.

First, machine learning (ML) techniques may be exploited [13–15]. That is, process executions (including their data) are encoded as a feature vector X. Common ML methods such as regression or decision trees are used to construct an estimator $\hat{q}(Y)$ conditioned on X. While such an approach is often accurate in predicting $q(Y)$, it has two major drawbacks. Given a performance measure Z that is not directly observable in the log, one cannot quantify $q(Z)$, since ML methods require labelled observations of $q(Z)$ in the training phase. For instance, Z may be the waiting time for a specific resource. If it is not recorded in the log, $q(Z)$ must be estimated. This estimation procedure may introduce an error, which reduces the accuracy of the learning technique. In addition, exploring to-be processes and 'what-if?' analysis of current process parameters is impossible due to lack of data that describes the effect of X on Y under the new terms.

A second approach to answer performance questions is to use operational process models. Given L, operational models such as GSPNs can automatically be discovered and enriched with performance information [16,17]. To quantify $q(Y)$, a corresponding query $q_M(Y)$ is evaluated over the model, e.g., with the help of simulation [17] or queueing theory approximations [5,12]. A model-based approach overcomes the aforementioned limitations. It supports queries for measures that were not directly recorded in the log and enables to-be performance analysis and sensitivity analysis (e.g., by changing the control-flow and altering activity durations).

The model-based approach suffers from a major drawback, namely *over-fitting* of the estimated $q(Y)$ with respect to L [12]. ML-based methods balance over-fitting of $\hat{q}$ to L by means of model selection, which comprises two main approaches, namely regularisation, and aggregation [18]. Specifically, the former focuses on *eliminating* the least significant parts of an ML model (e.g., pruning a regression trees, subset selection, Ridge regression and Lasso), while the latter aggregate parts of the model by projecting its feature space into a smaller state-space (e.g., principal component regression, and partial least squares). An analogy to ML model selection for performance process models, does not exist.

2.2. Balancing fitness and generalization in performance process models

We illustrate the need to balance between over-fitting and under-fitting (or generalization) using two models that were discovered from a real-world case in the healthcare domain (see our evaluation results for details). Fig. 1 depicts two process models, discovered using the Inductive Miner [19] with different noise thresholds: 0% for model (a) and 20% for model (b). We observe that noise filtering balances over- and under-fitting of the control-flow regarding the log, yielding less of a "spaghetti" model when filtering more events. However, the trade-off between over- and under-fitting is not addressed for the performance perspective. Moreover, the impact of filtering a specific percentage of traces from the event log on the goodness-of-fit of the resulting model is unclear.

As we later demonstrate empirically, a principled approach based on performance-driven model simplification, in turn, alleviates under-fitting in the performance dimension, thereby significantly improving the accuracy of the resulting models. This simplification can be executed by removing parts of the model or by aggregating its sub-parts into new components.

3. Performance analysis with generalised stochastic Petri nets

We start the section with syntax and semantics of GSPNs (Section 3.1), before turning to the use of GSPNs for process performance analysis (Section 3.2).

Please cite this article as: A. Senderovich et al., To aggregate or to eliminate? Optimal model simplification for improved process performance prediction, Information Systems (2018), https://doi.org/10.1016/j.is.2018.04.003