Model Repair by Incorporating Negative Instances In Process Enhancement

Master Thesis

Author : **Kefang Ding**

Supervisor : **Prof. Wil M.P. van der Aalst**
Prof. Thomas Rose
Dr. ir. Sebastiaan J. van Zelst

Registration date: 2018-11-15
Submission date: 2019-03-12

This work is submitted to the institute

PADS RWTH University

Acknowledgments

The acknowledgments and the people to thank go here, don't forget to include your project advice.

Abstract

Due to the digital transformation, process mining has enabled wider application in organizations and gained more attention from researchers. However, most of techniques are built on semi-supervised environment where only positive information is available. How to incorporate negative information into process mining is challenging.

This article provides a novel strategy on incorporating negative information on process enhancement. Firstly, based on the directly-follows relations of business activities, it builds a directly-follows graph by balancing the effect from the given existing reference process model, positive instances and negative instances of actual event log; Subsequently, the directly-follows graph is transformed into process tree and Petri net. At end, long-term dependency on Petri net is further analyzed and added to block negative instances on the execution, in order to provide a preciser model.

Experiments for our implementation are conducted into scientific platform of KNIME. The results show the ability of our methods to provide better model with comparison to selected process enhancement techniques.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Research in process mining has gained much interest over the past decade[8, 5]. The objective of process mining is to support the analysis of business process and provide valuable insights on processes. According to [8], techniques of process mining are divided into three categories: process discovery, conformance checking and process enhancement. Process discovery techniques focus on deriving process models from event logs of the information system, aiming to improve the understanding of real business process. Conformance checking analyzes the deviations between process models and observed behavior of its execution. Enhancement adapts and improves existing process models by extending the model with more data perspectives or repairing the existing model to better reflect observed behaviors.

Due to the increasing availability of detailed event logs of information systems, process mining techniques have recently enabled wider applications of process mining in organizations around the world[8]. After applying process discovery on the information system, a process model is fixed for organizations to guide the execution of business. However,in real life, business processes often encounter exceptional situations where it is necessary to execute process differing from the predefined model. The organizations need to adapt the existing process model to reflect the reality. Basically, one can apply process discovery techniques again to obtain a new model from event log. Yet, the discovered model tend to have less similarity with the original model[3]. As shown in [3], there is a need to change an existing model similar to the original model while replaying the current process execution. Here comes the model repair.

Model repair belongs to process enhancement and stands between process discovery and conformance checking. It analyzes the workflow deviations between event log and process model, and fix the deviations by adding sub processes on the model. As known, business in organizations is goal-oriented and aims to have high performance according to a set of Key Performance Indicators(KPIs), for example, average conversion time for the sales, payment error rate for the finance. However, there are little research on applying the process mining with consideration of performance[5]. It points the contribution of [2] to combine performance into process mining. In [2], the event log is firstly divided into positive instances and negative instances with respect to specific KPIs. Then, the positive instances are used to repair the model. In this way, negative instances are simply ignored, which results in a model with less precision.

As an example, there is an model presented in Figure 1.1 (a), where A is followed directly by B. During its execution in real life, an event log is generated:

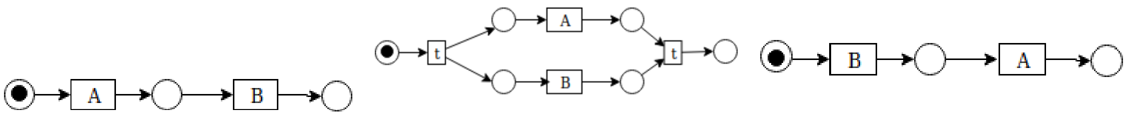
$$< A, B >^{55}, < B, A >^{105}$$

When considering KPIs performance, the log is divided into a positive and a negative set:

$$Positive\ examples : < A, B >^{5}, < B, A >^{100}$$

$$Negative\ examples : < A, B >^{50}, < B, A >^{5}$$

After applying current model repair techniques in [4], the process model is repaired using all examples. A can be skipped and duplicated later in a self-loop. In [2] methods, only the positive examples are taken for the model repair. Yet, since both $< A, B >$ *and* $< B, A >$ contributes to good performance, the repaired model keeps the same like in Figure 1.1(b).



(a) original process model

(b) process model with high fitness

(c) process model with good KPI

Figure 1.1: example for model change under model repair

However, it's obvious that $< A, B >$ often leads to bad performance and therefore should be excluded. The Figure 1.1(c) shows the expected model with incorporating the negative information. This model reinforces positive examples and avoids negative instances, which provides us a more accurate view of the business process.

The demand to repair model with incorporating negative instances appears. The concrete problem showing with demand is as follows. Given an input of one existing process model M, an event log L and KPIs, how to improve current process enhancement techniques by incorporating negative information within process model repair, and generate a process model to enforce the positive instances while blocking the negative instance. Therefore, the repaired model provides a better way to understand and execute the real business process.

This paper tries to provide a solution for it. The reminder is organized in the following order. Section 2 recalls the basic notions on process mining and list the preliminary to solve the problem. The next section lists our methods are introduced and formal definitions are given. In Implementation Section, the details of algorithms are given. Later, we evaluate our methods with simulated data and real data respectively and list the results. Subsequently, the discussion on this paper is presented. At last section, a conclusion is drawn on the paper.

Clearly, the use of negative information can bring significant benefits, e.g, enable a controlled generalization of a process model: the patterns to generalize should never include negative instances.

Chapter 2

Background

Process enhancement focuses to extend or improve an existing process model by using actual event log[8]. Besides extending the model with more data perspectives, repair is another type of enhancement. It modifies the model to reflect observed behavior while keeping the model as similar as possible to original model.

In [3], model repair is firstly introduced into process enhancement. By using conformance checking, the deviations of alignment are detected. The consecutive log moves is collected in the form of subtraces at specific location Q. Later, the subtraces are grouped into sublog that share the same location Q for subprocess discovery. In the earlier version in [3], the sublogs are obtained in a greedy way, while in [4], sublogs are gathered by using ILP miner to guarantee the fitness. Additional Subprocesses and loops are introduced into the existing model to ensure the fitness, which also brings variants of execution paths into the model.

Later, compared to [3, 4], where all deviations are incorporated in model repair, [2] considers model performance into model repair. An observation instance is built to represent the type of log moves given trace and KPI. Subsequently, a classification tree will be constructed from the set of observation instances to cluster traces of event log. Then, the techniques in [4] are applied into event log with positive traces to repair model.

As described above, the state-of-the-art repair techniques are based only on positive instances, meanwhile the negative instances are neglected. Without negative instances, it is difficult to balance the fitness and precision of those model. Few researches give a try on incorporating negative information into process discovery. [6] analyzes the available events set before and after one position and generates artificial negative events based on the complement of those event sets. Next, Inductive Logic Programming is applied to detect the preconditions for each activity. Those preconditions are then converted to petri net after applying a pruning and post-process step.

Similar work on model discovery based on negative information are published later. In [9], the author improves the method by assigning weights on artificial events with respect to unmatching window, in order to offer generalization on model.

[7] extends the techniques of numerical abstract domains and Satisfiability Modulo Theories(SMT) used in [1] to incorporate negative information for model discovery. Each trace in the log is treated as positive or negative and later transformed as one point in n-dimensional space, n is the number of distinct activities. The execution of a trace reflects the token transmission and marking limits on places in the model. Those limits are represented into the a set of marking inequalities and in a form of convex polyhedron

in n-dimensional space. Given half-space hypotheses, SMT solves the inequalities and gives the limits on the process model. Before SMT, negative information is incorporated to shift and rotate the polyhedron, which limits the generalization of the solution space. Because half-space is used, this method can not deal with negative instances overlapped into positive instances.

However, the field of model repair which considers the negative information is new.

To incorporate the negative information, simulated data are used, to limit the choices of going..

Compared to this, our approach is innovative mainly in the following aspects.

- Incorporate the negative information into model repair. Unlike the methods mentioned before
- Analyze the long-term dependency in the model to provide a preciser result.
- Analyze Model on Trace level. All activities constituting a trace are considered.

Bibliography

[1] Josep Carmona and Jordi Cortadella. Process discovery algorithms using numerical abstract domains. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):3064–3076, 2014.

[2] Marcus Dees, Massimiliano de Leoni, and Felix Mannhardt. Enhancing process models to improve business performance: a methodology and case studies. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 232–251. Springer, 2017.

[3] Dirk Fahland and Wil MP van der Aalst. Repairing process models to reflect reality. In *International Conference on Business Process Management*, pages 229–245. Springer, 2012.

[4] Dirk Fahland and Wil MP van der Aalst. Model repair—aligning process models to reality. *Information Systems*, 47:220–243, 2015.

[5] Mahdi Ghasemi and Daniel Amyot. Process mining in healthcare: a systematised literature review. 2016.

[6] Stijn Goedertier, David Martens, Jan Vanthienen, and Bart Baesens. Robust process discovery with artificial negative events. *Journal of Machine Learning Research*, 10(Jun):1305–1340, 2009.

[7] Hernan Ponce-de León, Josep Carmona, and Seppe KLM vanden Broucke. Incorporating negative information in process discovery. In *International Conference on Business Process Management*, pages 126–143. Springer, 2016.

[8] Wil Van Der Aalst. *Process mining: discovery, conformance and enhancement of business processes*, volume 2. Springer, 2011.

[9] Seppe KLM vanden Broucke, Jochen De Weerdt, Jan Vanthienen, and Bart Baesens. Determining process model precision and generalization with weighted artificial negative events. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1877–1889, 2014.