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. Sebastiaan J. van Zelst

Registration date : 2018-11-15

Submission date : 2019-??-??

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 the techniques are built on semi-supervised environment where only positive information is available. Therefore, it is challenging to incorporate negative information into process mining, in order to benefit the process model discovery or process enhancement.

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 and negative instances of actual event log; Subsequently, the directly-follows graph is transformed into process tree and Petri net by using Inductive Miner. 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

Process mining is a relatively new discipline that has emerged from the need to bridge data mining and business process management. The objective of process mining is to support the analysis of business process, provide valuable insights on processes and further improve the business execution. According to [12], 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, allowing the vision into the real business process. Conformance checking analyzes the deviations between an referenced process model and observed behaviors driven from its execution. Enhancement adapts and improves existing process models by extending the model with additional data perspectives or repairing the existing model to accurately 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[12]. After applying process discovery in organizations, a process model is fixed in information system 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. To reflect the reality, the organizations need to adapt the existing process model. Basically, one can apply process discovery techniques again to obtain a new model from event log. However, due to the facts, (1) the cost of rediscovery, and (2) the discovered model tend to have less similarity with the original model[6]. As shown in [6], 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 mainly 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 few researches on applying the process mining with consideration of performance[8]. [8] points out the rare contributions like [5] to combine performance into process mining. Deviations are firstly analyzed to determine if they have a positive impact on the process performance. Model repair techniques in [7] are applied into traces with positive deviations.

However, the current repair methods have some limits. Model repair fixes the model by adding subprocesses, silent transitions or loops, it guarantees the model fitness but overgeneralizes the model, such that it allows more behaviors than expected. On the other hand, it increases the model complexity. Even the performance is considered in [5], but only deviations in positive is used to add subprocesses, the negative information is ignored, which disables the possibility to block negative behaviors from model. An motivation example is listed to describe those limits.

1.1 Motivation Example

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:

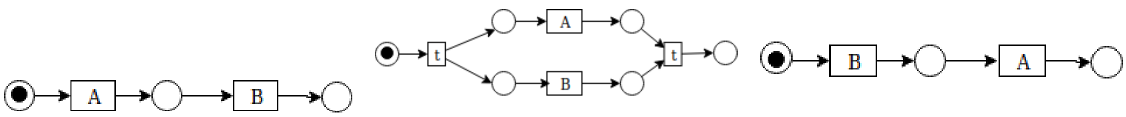
$$< A, B >^{55}, < B, A >^{105}$$

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 [7], the process model is repaired using all examples. A can be skipped and duplicated later in a self-loop. In [5] 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.

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. The demand to improve current repair model techniques with incorporating negative instances appears. In the next section, the demand is analyzed and defined in a formal way.

1.2 Research Problem Definition

We analyze the current model repair methods, and give the formal definitions.

Definition 1.1. Given an input of one existing process model M, an event log L and a set of KPIs, how to improve current process enhancement techniques by incorporating negative information, and generate a process model to enforce the positive instances while blocking the negative instance, with condition that the generated model should be as similar to the original model as possible Therefore, the repaired model provides a better way to understand and execute the real business process compared to the original model.

This paper tries to provide a solution for it. Our idea is to analyze the positive and negative impact on process performance of each trace. It balances the existing model, positive traces and negative traces on directly-follows relation, in order to incorporate all the factors on model generation. Later, the directly-follows relation is used to create process model by Inductive Miner. What's more, the impact of the existing model, positive and negative instances are parameterized by weights, to allow more flexibility of the generated model.

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.

Chapter 2

Related Work

To update an existing process model in organizations, there are two strategies, rediscovery and process enhancement. Process rediscovery applies the discovery techniques again on the actual event log. Process enhancement improves the model based on not only the actual event log but also the existing model.

Process discovery has been intensively researched in the past two decades and many algorithms have been proposed. They can be classified into the following categories, based on directly-follows relation, state-based regions, language-based regions are others. Directly-follows[13, 10] methods investigates the activities order in the traces and extracts higher relations which are used to build process models. State-based methods like [1, 3] builds a transition system to describe the event log, and then group the state regions into corresponding petri net node. Language-based algorithms uses integer linear system to represent the place constraint where the token at one place can never go negative. By solving the system, a petri net is created. Its representative techniques are Integer Linear Programming(ILP) Miner[14]. Other methods due to [15] include search-based algorithm like Genetic Algorithm Miner[4], heuristic-based algorithm Heuristics Miner[17].

Among those discovery methods, Inductive Miner is widely applied due to its effectiveness, simplicity, and soundness. Inductive Miner [10] guarantees to generate sound process models. A directly-follows graph are built according to the event log. It finds the most prominent split in event log, after the detecting the operators which include exclusive choice, sequence, parallelism and loop, the operators are used to build the process tree. Sublogs are created due to this operator and as inputs for the same procedure until single activities. Process tree is a block-structured model and can be easily transformed into Petri net.

When the actual event log differs a lot from the referred process model, it is suitable to use the rediscovery method to improve the business execution. However, in some cases, the process enhancement focuses to extend or improve an existing process model by using an actual event log[12]. 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 [6], 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 [6], the sublogs are obtained in a greedy way, while in [7], 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 [6, 7], where all deviations are incorporated in model repair, [5] considers the impact of negative information. In [5], the deviations of the model and event log are firstly analyzed, in order to find out which deviations enforces the positive performance. Given a trace and a selected KPI, an observation instance is built to correlate the number of each activity move with KPI output. Based on the observation instance, a set of rules are derived in form of decision tree. According to the rules, the original event log are divided into sublogs with traces matching the rules. Sublogs are then repaired to contain only trace deviations which have a positive KPI output. Following repair, the sublogs are merged as the input for model repair in [7]. According to the study case in [5], it provides better result than [7] on the aspect of performance.

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. [9] 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 [16], the author improves the method by assigning weights on artificial events with respect to unmatching window, in order to offer generalization on model.

[11] extends the techniques of numerical abstract domains and Satisfiability Modulo Theories(SMT) used in [2] 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. Our idea to incorporate negative information on trace level into model repair is innovative.

Bibliography

[1] Robin Bergenthum, Jörg Desel, Robert Lorenz, and Sebastian Mauser. Process mining based on regions of languages. In *International Conference on Business Process Management*, pages 375–383. Springer, 2007.

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

[3] Jordi Cortadella, Michael Kishinevsky, Luciano Lavagno, and Alex Yakovlev. Synthesizing petri nets from state-based models. In *Proceedings of IEEE International Conference on Computer Aided Design (ICCAD)*, pages 164–171. IEEE, 1995.

[4] Ana Karla A de Medeiros, Anton JMM Weijters, and Wil MP van der Aalst. Genetic process mining: an experimental evaluation. *Data Mining and Knowledge Discovery*, 14(2):245–304, 2007.

[5] 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.

[6] 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.

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

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

[9] 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.

[10] Sander JJ Leemans, Dirk Fahland, and Wil MP van der Aalst. Discovering block-structured process models from event logs-a constructive approach. In *International conference on applications and theory of Petri nets and concurrency*, pages 311–329. Springer, 2013.

[11] 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.

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

[13] Wil Van der Aalst, Ton Weijters, and Laura Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.

[14] Jan Martijn EM Van der Werf, Boudewijn F van Dongen, Cor AJ Hurkens, and Alexander Serebrenik. Process discovery using integer linear programming. In *International conference on applications and theory of petri nets*, pages 368–387. Springer, 2008.

[15] Boudewijn F Van Dongen, AK Alves De Medeiros, and Lijie Wen. Process mining: Overview and outlook of petri net discovery algorithms. In *Transactions on Petri Nets and Other Models of Concurrency II*, pages 225–242. Springer, 2009.

[16] 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.

[17] Anton JMM Weijters and Wil MP Van der Aalst. Rediscovering workflow models from event-based data using little thumb. *Integrated Computer-Aided Engineering*, 10(2):151–162, 2003.