
Model Repair by Incorporating Negative Instances In Process Enhancement

Master Thesis

Author : **Kefang Ding**

Supervisor : Dr. Sebastiaan J. van Zelst

Examiners : Prof. Wil M.P. van der Aalst
Prof. Thomas Rose

Registration date : 2018-11-15

Submission date : 2019-04-08

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

Process mining is based on business execution history in the form of event log, aim to bring visual insights on the business process and to support process analysis and enhancements. It bridges the gap between traditional business process management and advanced data analysis techniques like data mining and gains more interests and application in recent years.

Process enhancement, as one of the main focuses in process mining, improves the existing processes according to actual business execution in the form of event logs. The records in an event log can be classified as positive and negative according to predefined Key Performance Indicators, e.g. the throughput time, and cost. Most of the current enhancement techniques only consider positive instances from an event log to improve the model, while the value hidden in negative instances is simply neglected.

This thesis provides a novel strategy that considers not only the positive instances and the existing model but also incorporate negative information to enhance a business process. Those factors are balanced on directly-follows relations of activities and generate a process model. Subsequently, long-term dependencies of activities are detected and added to the model, in order to block negative instances and obtain a higher precision.

We validate the ability of our methods to incorporate negative information with synthetic data at first. Then, we conduct experiments in a scientific workflow platform KN-IME to show the statistical performance of our methods. The results showed that our method is able to overcome the shortcomings of the current repair techniques in some situations and repair models with a higher precision.

Chapter 1

Evaluation

This chapter presents an experimental evaluation of our techniques to repair model. At first, the evaluation measurements are defined. Next, we briefly introduce the test platform KNIME and ProM plugins tools for evaluation. In the following main part, the test on properties of our techniques is presented at the beginning. Then synthetic data is generated randomly to show the whole performance of our methods. At last, we conduct our experiments on real life data and also compare our techniques with other methods. The results show the ability of our techniques to repair model with high ranking according to defined measurements.

1.1 Evaluation Measurements

We evaluate our techniques based on the quality of repaired models with respect to the given event logs. In process mining, there are four quality dimensions generally used to compare the process models with event logs.

- *fitness*. It quantifies the extent of a model to reproduce the traces recorded in an event log which is used to build the model. Alignment-based fitness computation aligns as many events from trace with the model execution as possible.
- *precision*. It assesses the extent how the discovered model limits the completely unrelated behavior that doesn't show in the event log.
- *generalization*. It addresses the over-fitting problem when a model strictly matches to only seen behavior but is unable to generalize the example behavior seen in the event log.
- *simplicity*. This dimension captures the model complexity. According to Occam's razor principle, the model should be as simple as possible.

The four traditional quality criteria are proposed in semi-positive environment where only positive instances are available. Therefore, when it comes to the model performance, where negative instances are also possible. The measurement metrics should be adjusted. The repair techniques in this thesis are based on the labeled data and the repaired model can be seen as a binary prediction model where the positive instances are supported while the negative ones are rejected. The model evaluation becomes a classification evaluation. Confusion matrix has a long history to evaluate the performance of a classification model.

Table 1.1: Confusion Matrix

		repaired model	
		allowed behavior	not allowed behavior
actual data	positive instance	TP	FN
	negative instance	FP	TN

A confusion matrix is a table with columns to describe the prediction model and rows for actual classification on data. The repaired model can be seen a binary classifier and produces four outcomes- true positive, true negative, false positive and false negative shown in the Table 1.1.

- True Positive(TP): The execution allowed by the process model has an positive performance outcome.
- True Negative(TN): The negative instance is also blocked by the process model.
- False Positive(FP): The execution allowed by the process model has an negative performance outcome.
- False Negative(FN):The negative instance is enabled by the process model.

Various measurements can be derived from confusion matrix. According to our model, we choose the following ones as the potential measurements.

- recall. It represents the true positive rate and is calculated as the number of correct positive predictions divided by the total number of positives.

$$Recall = \frac{TP}{TP + FP}$$

- precision. It describes the ability of the repaired model to produce positive instances.

$$Precision = \frac{TP}{TP + FN}$$

- specificity. In opposite with recall, it measures the true negative rate.

$$Specificity = \frac{TN}{TN + FP}$$

- accuracy. It is the proportion of true result among the total number. It measures in our case how well a model correctly allows the positive instances or disallows the negative instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- F-score is is the harmonic mean of precision and recall.

$$F_1 = \frac{2 * Recall * Precision}{Precision + Recall}$$

After experiments we can see that accuracy gaine more weights than precision.

Generally, there is a trade-off between the quality criteria. So the measurements are only used to compare specific aspects of our techniques.

1.2 Experiment Platform

Data simulation part Try to automatize and standardize experiments, in order to make this experiments more effective, and easier to repeat. So KNIME is selected as one of the test platform to perform tasks. Also, some existing evaluation plugins in ProM are used to evaluate on the other aspects.

1.2.1 PLG

Process Log Generator(PLG) is designed to generate random business processes models and event logs according by setting the complexity parameters. Complexity parameters for model generation in PLG include the maximum number of branches for activities relation parallel, or exclusive choices, maximum depth of nested pattern, the weights on relations, and so on. The parameters to generate an event log are, for example, number of traces, and noise on the workflow or missing rate of trace heads.

In this experiment, process models in Petri net are randomly created by inputting simple, normal and complex parameters settings. Event logs are also simulated according to different complexity level. The setting details is in Table??.

1.2.2 KNIME

KNIME Analytics Platform is open source software in Java to help researcher analyze data by integration of multiple modules for loading, process and transformation and machine learning algorithms. Researcher can achieve their goals by creating visual workflows composed of modules with an intuitive, drag and drop style graphical interface, rather than focusing on any particular application area. The reason to choose KNIME as our experiment platform is that KNIME supports automation of test workflow which is more efficient. However, to conduct experiments, the related modules of our algorithm have to be integrated into KNIME, which requires additional development effort. At end, process mining modules like event logs, basic process models reader and writer, basic logs manipulators, classic discovery algorithms, and model repair are imported into KNIME.

1.2.3 ProM Evaluation Plugins

Here we discuss the plugins to test other aspects of our methods.

1.3 Experiment Result

1.3.1 Test On Property

In this experiment, we aim to answer the questions: 1) How do our techniques incorporate the existing model, positive and negative information to repair model? 2) How do the techniques overcome the shortcomings of current state-of-the-art repair techniques? To answer the first question, we applied the repaired techniques on event logs with different relations of activities, such as sequence, parallel and loop, exclusive choice and investigated the effect from the existing model, positive and negative instances on the repaired model, by adjusting the corresponding weights. For the second question, we describe the situations where our techniques overcome the defects of other repair techniques and explain the reasons behind it.

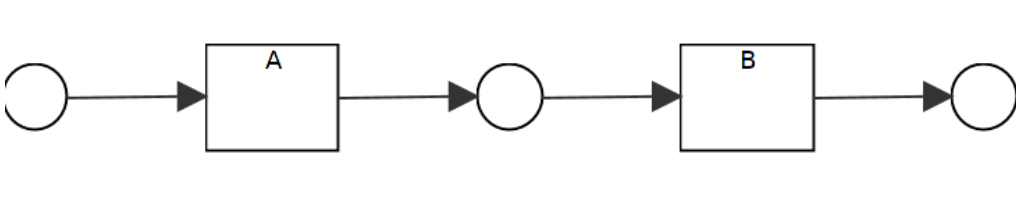


Figure 1.1: Model M1 with sequence relation

1.3.1.1 Test On Sequence

This part is used to show the effect of our techniques on the sequence relation of activities. Given a fixed model in Petri net with sequence relation, a set of event logs with different deviations are used to test if our repair techniques work properly. By changing the control weights of the existing model, positive and negative instances, different models are derived from the repair method. Next, we evaluate the model based on confusion matrix and plot the several measurements with the weight changes. By this way, we show how our method incorporates different effect of control weights.

In this experiment, we fix one weight as control variable while keeping the other weights unchanged. To make the experiments more reliable, for each fixed control variable value, we conduct the experiment in 5 different combinations of other weights that are randomly generated. At last, the average of those combination is used to represent the final measurement.

- *Experiment 1* – delete activity from sequence relation.
Event Log –

Positive : $\langle a \rangle^{50}$

Negative : $\langle a, b \rangle^{50}$

Result Analysis – Figure 1.2 shows the measurements results separately due to the weight change. In Figure 1.2a, with the weight for the existing model increases, precision, accuracy and specificity decrease while the recall keeps 1.0. After checking confusion matrix, we see TP and FN keep the same, FP goes up while TN goes down. The reason is that the higher weight for the existing model causes the higher possibility to keep the existing model during the repair. Due to the deviation of the existing model and actual event log, precision and accuracy goes down. With the weight for positive goes up, all measurements are stable in value 1.0. The reason for this is the TF and TN keeps the same, other values are all 0.0.

•

1.3.1.2 Test On Parallel

This part shows how the parallel relation of activities is affected by the weights for the existing model, positive and negative instances.

1.3.1.3 Test On Loop

This part investigated our repair method on activities with loop relation.

1.3.1.4 Test On Exclusive Choice

This part displays the changes of exclusive choices relation in the model under the different control weights.

1.3.2 Comparison To Other Techniques

This section represents some situations where current repair techniques can't handle properly, while our algorithm gives out an improved repaired model.

Situation 1, unfit part!! added subprocess are too much!! Where the addition of subprocesses and loops are allowed, while the structure changes are impossible, Fahrland's method applies the extension strategy to repair model by adding subprocesses and loops in the procedure. It introduces unseen behavior into the model. However, if the behaviors which are already in the model is unlikely to be removed from the model. One simple example is shown in the following part.

Dee's method is based on Fahrland's method. Deviations are calculated at first and used to build subprocesses for model repair. However, before building subprocesses, it classifies the deviations into positive and negative ones with consideration of trace performance. Only positive deviations are applied to repair model. Different to Fahrland's method, it improves the repaired model performance by limiting the introduced subprocesses. Still, it can't get rid of the defect mentioned before.

Situation 2, For fitted data in the model, can not recognize them!! where overlapped data noise can not be recognized, trace variant with more negative effect is treated as positive and kept in the model, which we should delete them.

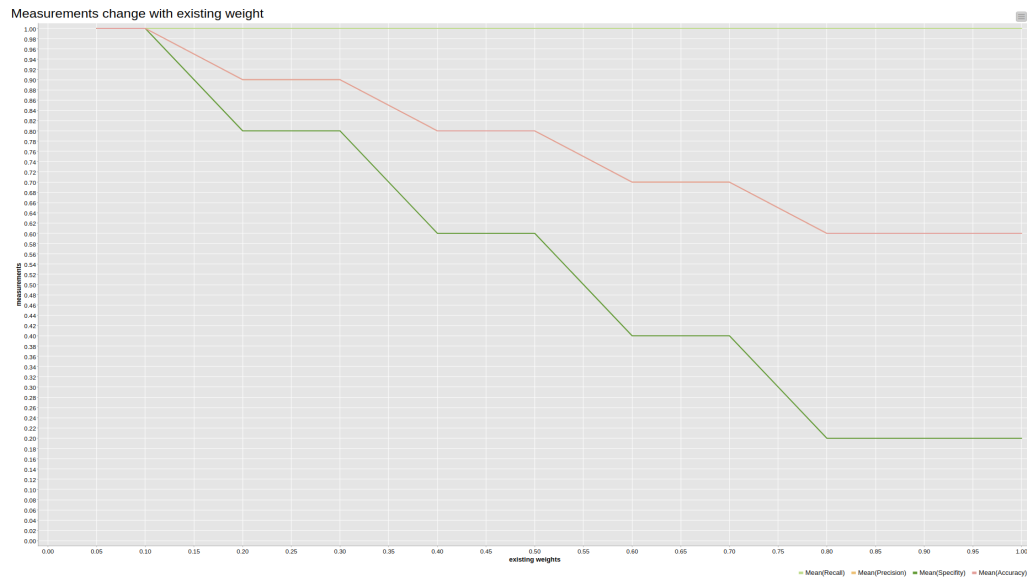
Situation 3, with long-term dependency!! fitted part or new added part!! none of the current techniques can handle this problem yet. Simple examples listed, but will this repeat the last section??

For one exclusive choices, but with long-term dependency detected and added in the model, precision and accuracy increase, since model with long-term dependency blocks the negative information by adding transitions and places to limit activity selection.

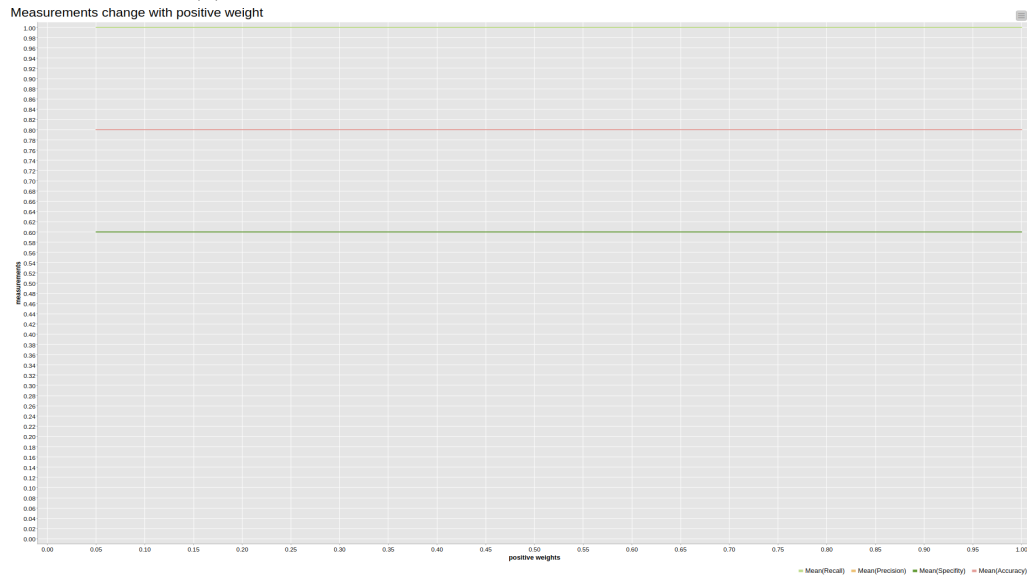
1.3.3 Test On Synthetic Data

Those synthetic data is generated randomly by using the simulation tool plg

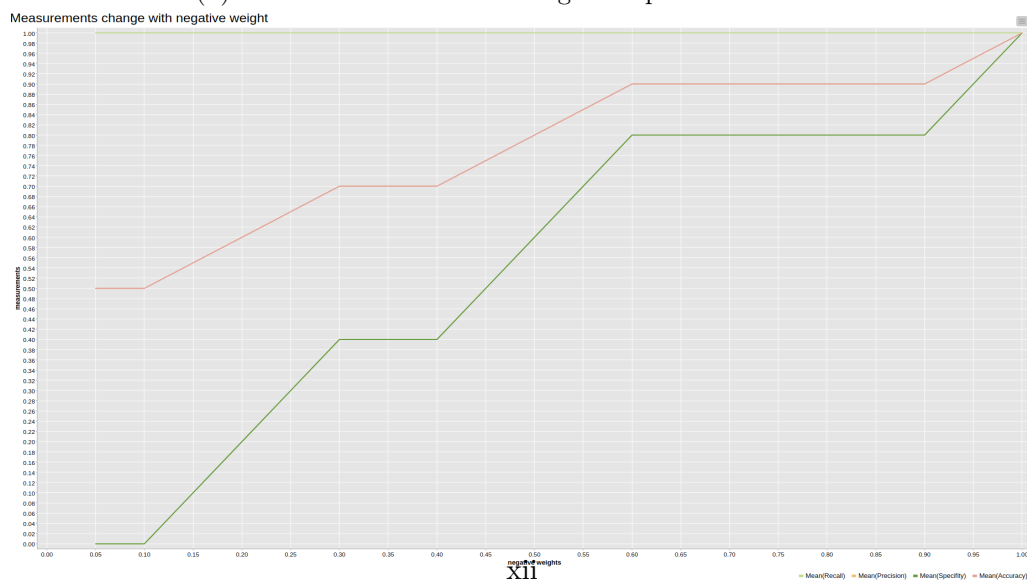
1.3.4 Test On Real life Data



(a) Measurements with the weight for the existing model



(b) Measurements with the weight for positive instances



(c) Measurements with the weight for negative instances

Figure 1.2: example for model change under model repair

Chapter 2

Conclusion

Bibliography

- [1] Wil Van Der Aalst. *Process mining: discovery, conformance and enhancement of business processes*, volume 2. Springer, 2011.
- [2] 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.
- [3] Mahdi Ghasemi and Daniel Amyot. From event logs to goals: a systematic literature review of goal-oriented process mining. *Requirements Engineering*, pages 1–27, 2019.
- [4] 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.
- [5] Dirk Fahland and Wil MP van der Aalst. Model repair—aligning process models to reality. *Information Systems*, 47:220–243, 2015.
- [6] Wil Van der Aalst. Data science in action. In *Process Mining*, pages 3–23. Springer, 2016.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.

-
- [12] 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.
 - [13] 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.
 - [14] 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.
 - [15] 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.
 - [16] Seppe KLM vanden Broucke, Jochen De Weerd, 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] 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.
 - [18] Josep Carmona and Jordi Cortadella. Process discovery algorithms using numerical abstract domains. *IEEE Transactions on Knowledge and Data Engineering*, 26(12): 3064–3076, 2014.
 - [19] BF Van Dongen, Jan Mendling, and WMP Van Der Aalst. Structural patterns for soundness of business process models. In *2006 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC’06)*, pages 116–128. IEEE, 2006.
 - [20] Wil van der Aalst. *Process Mining: Data Science in Action*. Springer Publishing Company, Incorporated, 2nd edition, 2016. ISBN 3662498502, 9783662498507.
 - [21] Joos C. A. M. Buijs, Boudewijn F. van Dongen, and Wil M. P. van der Aalst. On the role of fitness, precision, generalization and simplicity in process discovery. In *OTM Conferences*, 2012.
 - [22] Eindhoven Technical University. © 2010. Process Mining Group. Prom introduction. URL <http://www.promtools.org/doku.php>.
 - [23] Kefang Ding. Incorporatenegativeinformation. URL <http://ais-hudson.win.tue.nl:8080/job/IncorporateNegativeInformation/>.