Master Thesis – Math Formalization

Kefang Ding

9 Nov 2018

Abstract

Model repair in process mining aims to improve existing process model according to actual event log. Event log is divided into positive and negative instances based on given KPIs. However, the current repair technologies use only positive instances, while negative instances are ignored. This results in a less precision model. This article focuses on incorporating both positive and negative instances to repair model, in order to provide a model with better precision.

Firstly, a directly-follows graph is created from an existing process model in form of Petri net, positive and negative instances of event log. Then, the directly-follows graph is transferred to a process tree, which is used to generate the final model in Petri net. To improve the precision of Petri net, long-term dependency is analyzed and added to Petri net.

In comparison to current technologies, the methods proposed in this articles provide better result with respect on precision in most cases.

1 Introduction

Process Mining supports the analysis of business processes on three fields: process discovery, conformance checking and process enhancement[1]. 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 process models according to the data of real process.

Especially, process enhancement gains more and more interests from companies. With the application of process mining

Section description. The next sections are organized in the following order. Firstly, related work is given and compared. Secondly, 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. At last, a conclusion on the algorithms is provided.

It is about 1 page!!

2 Formal Definition

Copy the existing document and paste here..

I did some introduction, it is proven that in some cases, we can't deal with it. When the $XORB\_S \neq LT\_S || XORB\_T \neq LT\_T$.

Another way to avoid this problem is to add duplicated events, but the problem stays the same, so if we want to keep the model fit, we add new event for the discovered long-term dependency, the original, we keep it in the model?? But it is not precise!!! It allows too many choices there, but the model is sound, because we add events on model, we produce and consume tokens from duplicated events, and consumes it later.. Still, it is not so right. But if we choose the events before, we decides the events later, it's true... helps a little...

How to prove it ?? By induction. We use process trees as one internal result in our approach in two factors: (1) they are sound by construction, and can be transferred into sound Petri net models; (2) they are block-structured. which benefits the detection of exclusive relations in model.

- the original model is sound
- after adding one long-term dependency by duplicating the event, we make sure
  duplicated events are added into the xor branches, if it's chosen, then it consumes one token, at this xor branches, then it produces two tokens, one of which is put back again into the

  To prove the added places and duplicated events between two xor branches will not violate the soundness of model.

-
  - adding duplicated events of one long-term dependency does not violate the soundness of model.
  - To connect the source and target of long-term dependency which are the duplicated events will not violate the soundness.

    One extra place is added to connect the source and target. After executing the source, one token is generated in this place; Due to long-term dependency, only this target is triggered by this token and it consumes this token. No extra token is introduced into this model, so the model keeps sound.
- While keeping the original event in the model, the model is with less precision.

  So I want to solve this problem, and make it preciser by considering the original events into model... One drawback exists there, still... If we use the old and then the

It's about 12 pages. But one problem, some basic information, we forget to give .. Maybe, we can put the related work later!!!

3 Implementation

We can give some screen shoots and describes the work here..

For the implementation, we need to consider the correctness of our methods.

More details here are around 6 pages

4 Evaluation

Some tests are given, one is simulated data, and also some examples are given to explain the advantages. Later, real data is listed here..

It's about 3 pages or 4 pages with graphs.

5 Related Work

Related work, we give examples of new-generated model of Inductive Miner, Alpha Miner, ¡In which cases, it gives better results. ¿ Then, repair model, Fahland's methods, Anna's methods,

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.

It is about 2 pages!!

6 Conclusion

One page is enough.

References

[1] W. Van Der Aalst, *Process mining: discovery, conformance and enhancement of business processes*, vol. 2. Springer, 2011.