Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik

Resolving Partially Ordered Traces Using Deep Learning

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Science (B. Sc.)

eingereicht von: Glenn Dittmann
geboren am: 13.06.1993
geboren in: Berlin

Gutachter/innen: Prof. Dr. Matthias Weidlich
Prof. Dr. Han van der Aa

eingereicht am: ..... verteidigt am: .....

Contents

Purpose and scope of your entire report. The purpose of your entire report is to make a **scientific argument using the scientific method**.A scientific argument always has the following steps that all must come in this order.

SM1 **Explicate the assumptions and state of the art** on which you are going to conduct your research to investigate your research problem / test the hypothesis.

SM2 Clearly and precisely **formulate a research problem or hypothesis**.

SM3 **Describe the (research) method** that you followed to investigate the problem / to test the hypothesis in a way that **allows someone else to reproduce your steps**. The method must include steps and criteria for evaluating whether you answered your question successfully or not.

SM4 **Provide execution details** on how you followed the method in the given, specific situation.

SM5 **Report your results** by describing and summarizing your measurements. You must not interpret your results.

SM6 **Now interpret your results** by contextualizing your measurements and drawing conclusion that lead to answering your research problem or defining further follow-up research problems.

1 Introduction

Purpose and scope of Section 1. The introduction is a summary of your work and your scientific argument that shall be understandable to anyone in your scientific field, e.g., anyone in Data Science. A reader must be able to comprehend the problem, method, relevant execution details, results, and their interpretation by reading the introduction and the introduction alone. Section 1.1 introduces the general topic of your research Section 1.2 discusses the state of the art and identifies a research. Section 1.3 then states the research problem to investigate. Section 1.4 explains the research method that was followed, possibly with execution details. Section 1.5 then presents the results and their interpretation. Only if a reader thinks they are not convinced or they need more details to reproduce your study, they shall have to read further. The individual chapters and sections provide the details for each of the steps in your scientific argument.

You usually write the introduction chapter *after* you wrote all other chapters, but you should keep on making notes for each of the subsections as you write the later chapters.

1.1 Context and Topic (SM1)

1.2 State of the Art (SM1)

1.3 Research Question (SM2)

1.4 Method or Approach (SM3, SM4)

1.5 Findings (SM5, SM6)

2 Background

2.1 Preliminaries

Process Mining is a field in the area of data science, next to other famous domains such as statistics, datamining or machine learning. [Aal16, p.3ff] It's goal is to evaluate event data against some formal model, e.g. Petri Nets or BPMN models, describing the execution of processes. Event data is gathered manually or automatically likewise are process models. They can either be created according to given ideas and regularities describing the possible executions of an instance of the process. On the other hand there exist algorithmic techniques to discover process models from event data. With the given data and models it is then possible to evaluate certain properties of this data-model-complex, the three main types being discovery, enhancement and conformance. In the field of process mining, a subfield of (see process mining book) we deliberately investigate the happenings of processes of the "real world". It is about modeling, checking and improving such processes. Furthermore we can apply the area of conformance, checking on the area of process mining, which is basically a more in depth investigation how well a model of a process and the actual execution of such processes are conforming. This chapter is therefore used to gently introduce typical concepts that are used in process mining and more precisely defining the attributes we need in order to answer the research question. Additionally our (main) method to solve the problem of finding resolutions for partially ordered traces will be using neural networks, i.e. a special method from the wide field of machine learning. Thus we will give a short overview over typical machine learning ideas and more concisely introduce artificial neural networks, all of their properties and capabilities as well as the different network architectures we will be have used.
Most of the plain mathematical concepts should be of knowledge to the reader, and therefore instead of defined mostly explained in text. There might be some exceptions relating to the fields of either conformance checking or neural network such as conformance checking measures or algorithms, gradient descent functions for error minimizing in ANN's or the way of encoding input data in an ANN.
Informally an *Event Log* is a data structure holding all sequentially executed *activities* as *events* of a given *process model*. Thus for any *Event Log* $\mathcal{L}$ the universe of *events*, *activities* or *cases* are denoted as $\mathcal{E}$, $\mathcal{A}$ and $\mathcal{C}$ respectively.
For a running example we will refer to the b12-Log, which we also examined for solving the proposed research questions. This log is accessible at ??.

Definition 1. (Event) Let $\mathcal{E}$ be the universe of events. An event $e \in \mathcal{E}$ describes that a most single unit of a process has been executed. Events hold certain attributes, from the universe of attributes $\mathcal{N}$. So we define a function $\#_n(e)$ for an $n \in \mathcal{N}$, which describes the value of attribute n for event e.

Definition 2. (Activity) Let $\mathcal{A}$ be the universe of activities. An event $e \in \mathcal{E}$ is mapped to a distinct activity, i.e. $\#_{Acitivity}(e) \in \mathcal{A}$, which describes the unique activity that was executed for that event.

Definition 3. (Case) Let $\mathcal{C}$ be the universe of cases, then $\#_{Case}(e) \in \mathcal{C}$ is used for relating events to the same execution instance of a process.

The definition of cases is needed for defining *traces* later on.
In our case, i.e. in the logs we investigated, events hold various attributes, but only the attributes *ID*, *Acitivity* and *Time* are relevant. For instance $\#_{Time}(e)$ for an $e \in \mathcal{E}$ returns the exact time at which the happening of that was record. Note that this obviously must not be the time at which the event had happened really.

The classical definition of traces in process mining is that a trace T is a set of events $e_1, e_2, ..., e_n$ all related to the same case, i.e. for all $e, e' \in T, \#Case(e) = \#_{Case}(e')$. A *trace* E can also be represented as a sequence of activities $\sigma = \langle a_1, a_2, ..., a_{|E|} \rangle \in \mathcal{A}^*$, where $a_i = \#_{Activity}(e_i)$ of the sequence $\langle e_1, e_2, ..., e_{|E|} \rangle \in E^*$, which is obtained by ordering each event by its timestamps, i.e. for all $1 \leq i \leq j \leq |E|$ it holds that $\#_{Time}(e_i) < \#_{Time}(e_j)$. Since we consider events with equal timestamps and thus such ordering would be ambiguous we will define traces as sets of events. Each event set contains events with the same timestamp and thus we can order the log clearly.

Definition 4. (Trace) A sequence of disjoint set of events $E_1, E_2, ..., E_n \subseteq \mathcal{E}$ is a trace $\sigma = \langle E_1, ..., E_n \rangle$, if for all $e, e' \in E_\sigma$, $\#Case(e) = \#_{Case}(e')$, where $E_\sigma = \bigcup_{1 \leq i \leq n} E_i$ is the set of all events of σ and $\forall E_i \in \sigma \forall e, e' \in E_i \#_{Time}(e) = \#_{Time}(e')$ and $\forall E_j, E_k \in \sigma \forall e_{jl} \in E_j and e_{km} \in E_k \#_{Time}(e_{jl}) < \#_{Time}(e_{km}), 1 \leq j \leq k \leq n, l \in |E_j|, m \in |E_k|$

Similarly the definition of an Event Log is slightly deviant than from the usual ones, as for example used in [Aal16] or [CvDSW18], where Logs are built from the powerset of the event universe $\mathcal{E}$

Definition 5. (Event Log) An Event Log is a tuple (Σ, λ), where Σ denotes a set of traces and $\lambda : \bigcup_{\sigma \in \Sigma} E_\sigma \to \mathcal{A}$ assigns activities to all events.

Based on the timestamps associated with events in a given log, we define a partial ordering of events for a certain trace. Thus an event e1 < e2, if the timestamp of e1, i.e. time(e1), is smaller than/happens before, the timestamp of e2, i.e time(e2). Otherwise the timestamps of e1 and e2 are the same, leading to e1 = e2, i.e. time(e1) = time(e2). So for any Event Set $E = \{e_1, ..., e_n\} \in \wp(\mathcal{E})$there exist n! possible resolutions of that Event Set, i.e. orderings in which way those events could have been executed in real life. This is formally captured in the following definition.

Definition 6. (Possible Resolution) For an Event Set $E \in \mathcal{E}$ we define the possible resolutions of E as all ordered sequences of the elements in E with $\Phi(E) = \{\langle e_1, ..., e_{|E|}\rangle | \forall 1 \leq i, j \leq |E| : e_i, e_j \in E \wedge e_i = e_j \implies i = j\}$

For speaking of traces and logs according to their traits in respect to their certainty we define the notion of certain and uncertain traces or logs respectively, which resemble the appearance of uncertain event sets in either of the two.

Definition 7. (Certain, Uncertain Trace) A trace $\sigma = \langle E_1, ..., E_n\rangle$ is called certain if $\forall E_i \in \sigma : |E_i| = 1, i \in \{1, ..., n\}$. Otherwise σ is called uncertain

Definition 8. (Certain, Uncertain Event Log) Similarly an Event Log L is called uncertain if there exists an uncertain trace $\sigma \in L$, otherwise L is called certain.

Definition 9. (Artifical Neural Network) Artifial Neural Network (ANN)

Definition 10. (feedforward-ANN) feedforward-ANN

Definition 11. (Recurrent Neural Network) Recurrent Neural Network (RNN)

In the field of deep learning it is typically needed to encode your data for feeding it into your deep learning models such that the underlying algorithms can process and calculate the given input data. In our case the data is in a text context as events are mostly described in names.

Definition 12. (Encoding An encoding is a mapping of given data to some numerical value.

We will use classic encoding strategies for our first encoding approach (One-Hot- and Embed-Encoding) and use a second approach use an advanced version of the classic one-hot-encoding.

Partial Order, Total Order

2.2 Related Work

The field of conformance checking and process mining is very broad, so a lot of research has been done there up to today. Furthermore the field of machine learning has reached another peak of high interest in business applications, as well as media, teaching and research. The particular problem of solving partially ordered event logs from real-life process applications though, has been investigated in a fairly small amount compared to the above. Different techniques have been used to solve the problem differently and machine learning was only used once for predicting information of event logs.
To my best knowledge and also stated in previous work [LMFvdA14] so far there has been little research done addressing the problem of only partially ordered event logs.
M. de Leoni et al. presented a technique to abstract the problem of aligning partially ordered traces into a PDDL-encoded planning problem. This approach will either report,

that there exists no solution, i.e. optimal alignment, for an explicit trace and petri net. Or it will, in finite time, find an optimal alignment for a trace and petri net, whether or not the trace is sequential, i.e. totally ordered or a trace containing concurrent events, i.e. partially ordered. (note that by definition every totally ordered trace it also a partially ordered trace).[dLLM18]
Van der Aalst et al. took another approach in defining partially ordered traces and from those compute partially ordered alignments, with the aim to provide a model that can express concurrently running events and from there getting insight in the meaning of those. [LFvdA14] They researched the usefulness of those partially-ordered alignments with case studies relying on real-world data from a Dutch hospital. [LMFvdA14]
In [TVLRD17] Niek Tax et. al. introduce an approach to tackle three question not yet visited in the field of process mining. They use RNN's, LSTM's specifically, to answer three questions 1),2),3). However they let the order certainity of traces unvisited and thus, while exploring an answer for the next activity, omit the information of certain parts of a trace already being order / they only give answer on how future traces could end most possibly.
Weidlich et. al. have sought three algorithmic approaches for resolving partially ordered traces in giving a probability distribution over all possible resolutions and from there on efficiently compute the conformance of partially ordered traces with a given process model.[vdALW19]
As of my best knowledge no research has yet been done, to resolve partial ordering of traces in conformance checking / process mining. We expect to find valuable solutions for resolving the partial ordering of events happening at the same time, exploiting the field of deep learning, neural networks respectively.
Trying to cite the books read so far:
Process Mining [Aal16]
Conformance Checking [CvDSW18]
Deep Learning [GBC16]
Hands-on ML [Gér19]

3 Problem Exposition (optional)

3.1 Context / Business Understanding (SM1)

As stated in the the Introduction 1 when logs are acquired in real life not all data will always be completely clean. For instance roles performing a certain activity might be missing or trace, i.e. instances of a process, might not have been recorded. Furthermore there is are multiple reasons for timestamps in the log to be vague. This can be due to the lack of synchronization, manual recording or data sensing. [vdALW19] For conformance checking techniques to be used on the even data to be executed properly and thus give meaningful insights into the correlation of model (Petri Net, ...) and data (Event Logs) there are different approaches one could undertake. For instance one could just drop out the uncertain traces before applying state of the art conformance checking

techniques, which especially makes sense when there is only a small fraction of traces that are uncertain for the event log.
However we would rather try to resolve this uncertainty to get the most realistic image of what has happened when the activities in the logs had been executed.

3.2 Data Understanding (SM1)

To understand the data we are working with better we examined x Event Logs available from the open research website from the 4TU website. (link is now down unfortunately). Event Logs that are generated in real world processes vary largely in a lot of different aspects. For example for the x logs we examined a priori, to choose the right ones for evaluating the research question upon, the number of traces range widely from about 1000 to over 150000 per Event Log and furthermore the size of the Activity Universe $\mathcal{A}$ can be quite small with 9 activities or relatively large with up to 51 activities executable.
SEE TABLE 1 FOR LOG SUMMARY
The uncertainty of the logs, i.e. the fraction of traces, which are uncertain, go from 6% up to almost the whole log being uncertain with 96%. The number of uncertain sets we observed appearing in the given logs go from 15 up to 39. The basic statistical key figures however are fairly evenly distributed over all the logs. So the average length of the uncertain sequences is between 2 and 3 and the median of the maximum length of the uncertain Sequences in each log is 4.

3.3 Detailed Research Questions (SM2)

The research question in detail tries to answer whether a dynamic deep learning neural network architecture can yield more precise and consistent results for partial order resolution in event logs than the statical algorithms approached in [vdALW19].
Therefore we need to ensure that we can define a model that suits our needs the best and the training data we will use for evaluating the deep learning model. The final goal of the research will be to acquire a model that we can feed any uncertain event set, for certain event set the problem is trivial, and it will present us the most probable resolution for that uncertain event set.

3.4 Detailed Method (SM3)

The method is to take real world event logs form open research repositories (link is down..) and process them as follows.
We extract all important data from the logs, mainly the traces and events. Additionally for each event we keep the needed attributes, which are the attendant Case-ID, Event-ID, timestamps and therefore their ordering. Also for comparism with the prior approach shown in [vdALW19] we implemented the three given algorithms in python and evaluated the results against what we could find with the deep learning approach of this thesis.

4 First Real Chapter addressing first Research Problem

4.1 Encoding the Event Data

Finding good encodings for events, is difficult, since all possible resolutions for an uncertain set leads to too many possible resolutions, i.e. the input or output vectors would get too large, when using *one-hot-encoding*. With *embeddings* the length of each input would be shorter or the chosen embedding dimension more precisely. But still the set of all possible resolution would be needed to calculate the embeddings for any occuring event set(maybe change name in definition) of the input log.
To address this problem only the occuring resoltions for any event set have been taken into account. The reduction of vector size for the five logs examined is represented in TABLE.
For any uncertain event set $e = e_1, e_2, e_3, ..., e_n$of length n, the size of the input vector results of following formula: FORMULA. The first part is the size of the activity universe, since all possible singular event sets must be considered.
GIVE FORMULA FOR INPUT VECTOR LENGTH
GIVE FORMULA FOR OUTPUT VECTOR LENGTH
This clearly increases exponentially by increasing the size of the activity universe or the length of the traces.
Encoding all possibilities vs encoding only the occurring possibilities
If we were to encode all possible up-to-k subsets for a given event set the number of encodings, i.e. the number of possibly appearing events is computed as follows. Thereby the number k is set to four at the moment, as the inspection of our three data sets / logs have shown the longest uncertain sequence is actually four.
$\#encodings(k = 4) = \sum_{j=1}^{k} |\mathcal{E}|^j$
For the three observed logs with a size of the event universe of 24 (BPI12), 9 (BPI14) and 11 (Traffic) respectively this would lead to a total number of 346200 (BPI12), 7380 (BPI14) and 16104 (Traffic) of possible events to encode.
However not all of these combinations of events do appear in the log. Actually we could observe that only 14 (BPI12). 24 (BPI14) and 25 (Traffic) of all possible event sets where present in the corresponding logs.
Summing up the possible resolutions for each of these event sets as of: let m be the length of an uncertain set, then there are m! possible resolutions for that set. E.g. for the uncertain set {a,b,c,} of length 3 there exist 3! = 6 possible resolutions, {a,b,c}, {a,c,b}, {b,a,c}, {b,c,a}, {c,a,b}, {c,b,a}. Counting up from there we have 86 (BPI12), 134 (BPI14) and 63 (Traffic) encodings for the output vectors. Leveraging this idea would on the one hand ease encoding in making the feature vectors a lot smaller. In the best case scenario for the BPI12 log this would mean a reduction of feature vector length by roughly the factor of 25000.
Furthermore one could reduce the size of the input vectors only(!), since ordering for any event set here does not matter / does not exist. So for instance the uncertain event set {a,b,c} and {c,a,b} resemble the same input structure and therefore could be treated as the same input, namely {a,b,c} in this case.

Additionally we only considered a k of 4. One however can easily see that for larger k this also explodes exponentially.
That is why embeddings where used for encoding, as literature suggests using this type of encoding over one-hot-encoding for vectors that would contain more than 50 different features.

Finally we consider two relevant encodings.

4.2 What is the training data

4.3 Trying out a feedforward-ANN

When considering the research problem, curiosity made the way for also trying out the learning behaviour of a feedforward-ANN. The hypothesis was that the feedforward-ANN would learn the current fraction of already correct ordered uncertain traces, when assuming the order in the log was the correct order for any given trace. By that way for a uncertain event say $\{a, b, c\}$ the net would possibly learn which ordering, $\{a, b, c\}$, $\{a, c, b\}$,$\{b, a, c\}$, $\{b, c, a\}$, $\{ca, b\}$, $\{c, b, a\}$ appears most frequently in the given log, and assign the each of the possible resolutions their following appearance probability.
´

5 Acquiring Data

5.1 Abstracting timestamps

5.2 Fraction of uncertain traces

5.3 Training and Test Data

5.4 Synthetical Data

5.5 Finding out the solutions and making predicitons

6 Evaluation

6.1 Objective (SM2)

6.2 Setup (SM3)

6.3 Execution (SM4)

6.4 Results (SM5)

6.5 Discussion (SM6)

7 Conclusion

Your conclusions are not just a factual summary of your work, but they position, interpret and defend your findings against the state of the art that you discussed in Sect. 1.2. You specifically outline which concrete findings or methodological contributions advance our knowledge towards the general objective you introduced in Sect. 1.1. Objectively discuss which parts you solved and in which parts you failed.
You should explicitly discuss limitations and shortcomings of your work and detail what kind of future studies are needed to overcome these limitations. Be specific in the sense that your arguments for future work should be based on concrete findings and insights you obtained in your report.

References

[Aal16] Wil van der Aalst. Process mining : data science in action / by wil van der aalst, 2016.

[CvDSW18] Josep Carmona, Boudewijn van Dongen, Andreas Solti, and Matthias Weidlich. *Conformance Checking*. Springer, 2018.

[dLLM18] Massimiliano de Leoni, Giacomo Lanciano, and Andrea Marrella. Aligning partially-ordered process-execution traces and models using automated planning. In *Twenty-Eighth International Conference on Automated Planning and Scheduling*, 2018.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[Gér19] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[LFvdA14] Xixi Lu, Dirk Fahland, and Wil MP van der Aalst. Conformance checking based on partially ordered event data. In *International conference on business process management*, pages 75–88. Springer, 2014.

[LMFvdA14] Xixi Lu, Ronny S Mans, Dirk Fahland, and Wil MP van der Aalst. Conformance checking in healthcare based on partially ordered event data. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–8. IEEE, 2014.

[TVLRD17] Niek Tax, Ilya Verenich, Marcello La Rosa, and Marlon Dumas. Predictive business process monitoring with lstm neural networks. In *International Conference on Advanced Information Systems Engineering*, pages 477–492. Springer, 2017.

[vdALW19] Han van der Aa, Henrik Leopold, and Matthias Weidlich. Partial order resolution of event logs for process conformance checking. 2019.

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und noch nicht für andere Prüfungen eingereicht habe. Sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, sind als solche kenntlich gemacht. Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Berlin, den 13/10/2020 .....