Quantitative estimates for recurrence rarity using a Markovian approach

van Cauter, Zeno M.

Award date:
2019

Quantitative estimates for recurrence rarity using a Markovian approach

ing. Zeno van Cauter (z.m.v.cauter@student.tue.nl)
Student of the Bachelor Data science, Technical University Eindhoven

August 20, 2019

Abstract

Process mining often revolves around conformance checking based on a predefined process model or process discovery based on event data. Many of these models contain structures that allow for recurrent behaviour within models. Within this paper it will be argued that current mainstream process models are inadequate for describing recurrent behaviour when probabilities for recurrences change. A proof of concept for an ideal model for this recurrent behaviour is described and possible suitable models are explored using illustrative examples. This paper also shows how to model the increasing recurrence probabilities in our example adequately and how to verify this using quantitative measures. In addition, the paper explores higher-order Markov chains for the modelling recurrence probabilities and uses literature-based techniques for determining the optimal order. The end result is a technique that will allow for a generalizing model that can provide quantitative estimates on how rare unobserved event sequences are.

Keywords: process mining, statistical modelling, petri net, process discovery; conformance checking; outlier detection;

Contents

Introduction

Process mining is a relatively new part in the field of data science and is used to bridge the gap between data science and process science (van der Aalst, 2016). The increasing awareness regarding the importance of event data has caused the field of process mining to surge and, like any other field of research, comes with researchers approaching the field from different angles or philosophies.

One of these views (Buijs, van Dongen, & van der Aalst, 2014) approaches it from the philosophy that there is a data generating system in place behind the actual observations in the event log, and a process model fit to the event log. These 3 parts: a data generating system, an event log and a process model, may each have a different approach depending on what the data scientist wants.

Regarding the topic of a data generating system (Buijs et al., 2014) can be quoted stating: *"It is important to realize that there is generally no way to explicitly describe the behaviour of the system, first of all since this behaviour is typically infinite, but more so because there is always the possibility of unforeseen behaviour in any realworld system that may even change over time"*.

One could see how good the event log fits on an a priori defined (and assumed or intended) process model (conformance checking), or see if process model based on the event log can be retrieved (process discovery) (van der Aalst, 2016). Examples of process discovery models are the Alpha Miner, Inductive miner and the Heuristics miner. Each of these algorithms will try to (given a predefined set of hyperparameters) fit process models in the best way possible to a given event-log.

However, like any other data science model, process discovery algorithms come with advantages and disadvantages in the form of which characteristics of the event log it reflects in the resulting model. The performance of a discovered process model is often measured through several basic measures (vanden Broucke, De Weerdt, Vanthienen, & Baesens, 2013). For model accuracy these are: recall (ability to replay the event log), precision (not overfitting the event log) and generalization (not underfitting the event log). For model comprehensibility these are: simplicity (Occam's Razor) and structuredness (ease of interpretation).

For general purposes of data science, a balance of precision and generalization is required. *"Although models should be precise, generalizing beyond observed behaviour is also a necessity. This is because assuming that all behaviour is included in an event log is a far too strong completeness assumption"* (vanden Broucke et al., 2013). Meaning that models *"should prevent the execution of unseen and unwanted behaviour"* but also *"allowing unseen but very likely (not explicitly forbidden) behaviour when taking into account the data in the event log"* (vanden Broucke et al., 2013).

One important characteristic affected by these two measures is the recurrences within an event-log. Here the definition of a recurrence entails seeing an activity multiple times within a single event sequence. Process discovery models will often include recurrence behaviour by allowing models to go back and forth between events. However, what this modelling does not account for, is that probabilities may shift for recurrences. This means that process discovery models can score high on precision and generalize poorly without knowing.

Another approach, although a more classic one, is to model event sequences as a

Markov chain model. Markov chains are a mathematical model designed to describe ”random walks” across ”states” and can produce event sequences similar to event-logs retrieved from IT-systems. Like process discovery algorithms, Markov chains also have specific advantages and disadvantages. However, because Markov chains have their origin in random walks, they can also produce event sequences that have no basis in the event log. Hence, Markov chains trained on event-logs tend to have poor precision but generalize well.

Like process discovery models, classical Markov chains also do not accomodate shifting probabilities for recurrences. Hence, this project will seek to resolve two problems. First, a literature study will be performed to properly define what needs to modelled to accomodate recurrences. Second, proposed models from the first problem be explored through data as a proof of concept for modelling recurrences with process models.

1 Problem definition and hypothesis

1.1 Method for obtaining the well-defined problem and proposing the hypothesis

First, the background of this topic will be explored to provide basis for why this examination is neccesary from a contextual point of view. Second, this project will have a look at what literature says about models that describe busines processes. The third step will specifically focus on using the literature to posing the problem in a short and well-defined manner and developing the research questions for the project.

1.2 Background

Many modern daily events are the result of underlying processes. If you would undergo medical treatment in the Netherlands, an example event sequence could look like this:

1. Visit to your physician/general practitioner
2. Get a referal to a specialist (often a hospital)
3. Get a scan
4. Get treatment in case the scan yielded results that indicate presence of a disorder
5. Get a scan to check if condition has been resolved
6. End with aftercare

This process, while simplified, displays an important characteristic of many processes: recurring events. The example event sequence has one recurring event: getting a scan. In real life, recurring events could happen much more frequently. An example is patients with cancer, who even after a treatment have to come back repeatedly for scans,

to check whether the disease has not relapsed. Recurring events could also consist of multiple events. Once diagnosed with the HPV-virus, you often will have to come back repeatedly for scans but also for a large amount of surgeries. Hence, repeating checks and surgeries often occur in sequence.

However, one may assume that patients and their treatment-process differ when looking at them at the large. It seems reasonable to assume that, in the context of hospital patients, not all patients have to come back for treatment the same amount of times. Perhaps probabilities for another recurrence change based on the context. For example, one would expect that most people to change their behaviour once arrested, meaning that the probability for another recurrence should be expected to decrease. An exception to this case would be revolving door criminals, where one would expect the probability to remain increasing or consistently high.

1.3 Problem and hypothesis

There are many examples where one would expect to not only see a distribution in the number of recurrences, but also changing probabilities as events are recurring. Yet, many commonly used process models or process discovery algorithms do not account for this characteristic. Development of a model that does incoorporate shifting recurrence probabilities can provide insight in event sequences from a different perspective than traditionally provided. To make an even stronger proposition, if it can be assumed that recurrence probabilities follow a pattern (a distribution if you will), a direct quantitative measure can be provided in telling how rare a so far unobserved event sequences would be. All in all, answering a set of questions is required to see if and how such recurrence probabilities model would work.

1. How do current mainstream process models work and how do they fail to incoorporate shifting recurrence probabilities?

2. What properties does a model require to be able to model recurrence probabilities?

3. Are there models that (possibly with little augmentations or alteratings) fit this description?

4. Using a proper model, can recurrence probabilities be modeled as a distribution? Can patterns be imposed onto these probabilities? How can these be verified?

5. How can a recurrence probability distribution to classify rarity of yet unobserved event sequences?

Question 1,2 and 3 will be tackeled as subproblem 1: *"Defining and developing the ideal model for recurrence probabilities"*. Questions 4 and 5 will be tackeled as subproblem 2a and 2b: *"Exploring models using data"*. The ultimate goal of this project is to show that current mainstream process models do not account for specific behaviour, and to provide a proof of concept on how to model this behaviour.

2 Subproblem 1: Defining and developing the ideal model for recurrence probabilities

2.1 Method

First, Petri nets and Markov chains will be explained, their differences highlighted and a reasoning will be provided for why these models are not sufficient for many practical purposes. This will be done using the ”Loan Application” dataset from the 4TU repository. It is a synthetic dataset selected for its simplicity as the data only consists of event sequences which can consist of 7 different possible events (called A to G). After that, specific properties of an ideal model will be proposed using a piece of probability theory.

2.2 Petri nets, Markov chains and their differences

A Petri net is a process model which makes use of transitions (squares) and places (circles). Places can hold tokens and transitions can move tokens around (or create/destroy them). An example is given in figure 1.

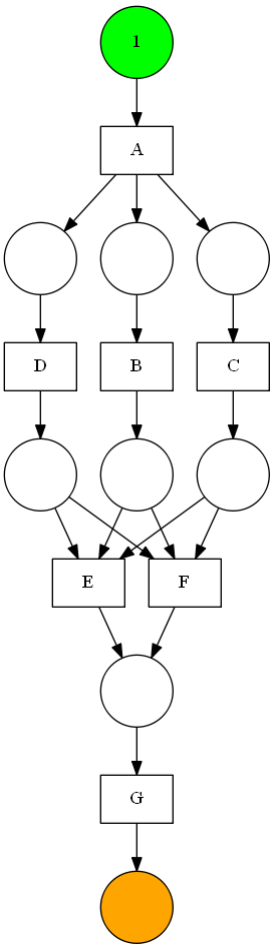

Figure 1: Resulting Petri net from loan application data.

The Petri net in figure 1 says that A must always be the first event and G the last 1. According to this model, any event sequence must start with $\langle A,B,C,D\rangle$ where B,C,D can happen in any order, and ends with either $\langle E,G\rangle$ or $\langle F,G\rangle$. So $\langle A,D,C,B,F,G\rangle$ would be a valid event sequence. Note how the Petri net tells us nothing on the probability of e.g. B occuring before C.

2.3 Markov chains

A discrete Markov chain is a piece of mathematics designed to describe ”random walks” across ”states”. Probabilities within a Markov chain are defined as the probability that it will move to a certain next state, given its current state. This is defined as the Markov property and is mathematically described as:

$$P(X_{n+1} = x|X_1 = x_1,..,X_n = x_n) = P(X_{n+1} = x|X_n = x_n) \quad (1)$$

The Markov property implies that the probability of the next state depends on the current state. Every entry in figure 1 can be seen as $P(X_{n+1}|X_n)$, the probability to move from state X_n to state X_{n+1}. In this example, the Markov matrix satisfies the Markov chain properties: the Markov chain has a well-defined state-space and every column sums up to 1.

to \ from	A	B	C	D	E	F	G
A	-	-	-	-	-	-	-
B	0.90	-	0.06	0.04	-	-	-
C	0.04	0.54	-	0.47	-	-	-
D	0.06	0.38	0.46	-	-	-	-
E	-	0.01	0.13	0.07	-	-	-
F	-	0.07	0.35	0.42	-	-	-
G	-	-	-	-	1.0	1.0	-

Table 1: Markov chain M for loan application data.

In table 1 the columns represent the starting state (X_n) and the rows represent the end state (X_{n+1}). E.g. starting in A has a 90% probability to go to B, 4% probability to go to C and a 6% probability to go to D. Note that the sum of a column occasionally exceeds 100% due to rounding-errors from formatting. Like the Petri net, going from A you can only go to B, C or D and from E and F you can only go to G. However, the big difference between the Petri net model and the Markov chain, lies in the relation between B,C and D. In the Petri net model B,C and D must all be executed, but the order is at will. Because of this parallel behaviour in the data, the trained Markov chain model reflects this by allow B to go C or D, C to B or D and D to B or C. This means that the Markov chain model contains recurrences that have no basis in the data but are reflected in this manner because of parallel behaviour.

From the Petri net it is known that $\langle B,C,D\rangle$ can only occur once and must occur in sequence (order at will). However, the mathematical model from the markov chain was originally designed to simulate "random walk" behaviour. The consequence of this is that if event sequences are modelled as Markovian behaviour, predicted event sequences may contain multiple B's, C's or D's which do not appear in the original event log. Technically, in the Markov model, sequences of $\langle B,C,D\rangle$ (order at will) may keep repeating due to its never decreasing probability. A visualization of the entire process and all its possible movements are given in figure 2.

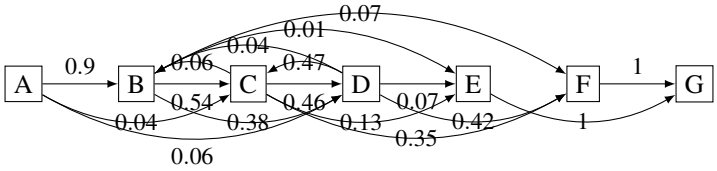

Figure 2: Visualization of the Markov chain.

The model in figure 2 allows for recurrent behaviour, even though this does not actually occur in the original dataset as visualized in the original Petri net (figure 1). Petri nets do allow for recurrences, but not with shifting probabilities. Markov chains allow for recurrences (even if they are not there) but as the Markov matrix is static, also does not allow for shifting probabilities. Before further progress can be made, specification is needed for what an ideal model should at least incorporate.

2.4 Ideal model for shifting recurrence probabilities

A problem arises when the fundamental underlying assumption of Markov chains needs to be dealt with: that the probability of the next step only depends on what state the process is currently in. From the process perspective, the probability p of seeing a certain event again may not be static and may depend on factors such as how many times the loop has already occured or other prior events in the event sequence.

For the first goal (describing the ideal model), the starting literature from Diogo R. Ferreira, who has published multiple papers on making Markov chains fit better to business processes (Ferreira, Szimanski, & Ralha, 2012). Unfortunately, the main literature ((Ferreira et al., 2012) and (Ferreira, Szimanski, & Ralha, 2014) quickly seize to provide proper insights, as both papers focus on proposing algorithms to map subsequences of events as high-level activities. This means that subsequences of a total event sequence are seen as seperate Markov chains. Hence, the overall process is modelled as (hierarchical) nested Markov chains. This literature is appropriate for when the recurrences are longer than a single event, but do not provide a proper setup for what is trying to be modelled in this project.

Just like traditional statistical modelling, the true data generating process cannot be observed and hence it may be unknown whether an observed event log contains all possible event (sub)sequences. (Ayo, Folorunso, & Ibharalu, 2017) attempts to generate models with 100% completeness using Fuzzy-Genetic Mining models based on Bayesian Scoring Functions. However, (Ayo et al., 2017) explicit focus lies on trying improve general scores (soundness and completeness) of traditional process models.

In general, while looking for already existing models, literature provided mostly models that were altered version of traditional concepts. For example, Hidden Markov Models (HMM) (Rabiner, 1989), Time Varying Markov Regime Switching Models (TVMRSM) (Bazzi, Blasques, Koopman, & Lucas, 2017) or High-order Markov chains (HOMC) (Raftery, 1985). Like (Ayo et al., 2017), HMM's are more focussed towards modelling the data generating process, similar to how Linear Regression models model a data generating process. A large variety of time focused Markov chains (e.g.

TVMRSM) were focussed more on explaining time-series and not suitable for repurposement.

For the second goal, literature (Conforti, Rosa, & t. Hofstede, 2017) was found which dealt with describing proposed technique that will allow near automated filtering of infrequent behaviour. This work makes use of ILP modelling and basic statistical descriptives (e.g. relative frequency, interquartile range) but does not look into the modelling these infrequent (sub)sequences as possible observations from statistical models.

Higher-order Markov chains initially seemed to provide more utilities compared to other models from literature. To explain this view, first a brief explanation of what HOMC's are and how they differ will be given.

While Andrey Markov originally came up with the concept of a Markov chain and the Markov property, subsequent work by mathematicians has lead to the generalization in the form of "High-order Markov chains". The order here represents how many previous states are "remembered" by the Markov matrix.

This means that our original default Markov property is a first-order Markov chain

$$P(X_{n+1} = x|X_1 = x_1, .., X_n = x_n) = P(X_{n+1} = x|X_n = x_n) \tag{2}$$

If the 2 previous states need to remembered, it will result in a second-order Markov chain

$$P(X_{n+1} = x|X_1 = x_1, .., X_n = x_n) = P(X_{n+1} = x|X_n = x_n, X_{n-1} = x_{n-1}) \tag{3}$$

and so forth. (Raftery, 1985) provides more in-depth on the derivation and specific properties of higher-order Markov models. Nonetheless, this will allow for more storage of (non-specific) memory into the model. Remember the first Markov chain based on the loan application data. Instead of getting for example probabilities $P(X_{n+1} = C|X_n = B)$, the probabilities $P(X_{n+1} = C|X_n = B, X_{n-1} = A)$ would be obtained.

Concluding, based on literature there does not appear to be an already developed model that perfectly fits the needs for this research. However, some models (classic Markov chains, higher-order Markov chains) may be suitable if they can be augmented or altered. To do this, definitions and properties need to be set in stone for what a recurrence means mathematically. This could be one of the models that have been seen previously, but altered or augmented. It is important to consider that when this is done, properties of previous models may not apply or a new model may require specific assumptions for it to work (e.g. the stationary distribution of the Markov chain may no longer exist or be consistent).

2.5 Required properties of ideal model

In a first-order Markov chain the Markov matrix is filled with entries representing $P(X_{n+1} = x|X_n = x_n)$, where x and x_n be selected from a set of states. However, recurrences manifest themselves differently. More specifically, the goal is to retrieve the probability of having another recurrence $(k+1)$ given that is has already seen a certain amount (k) recurrences. This means that the main objective is to model (similar to the Markov chain notation), $P(K = k+1|K \geq k)$.

In essence, this means that instead of the classical view of seeing the event as the state-space $\mathcal{S} : \{S_1 : A, S_2 : B, .., S_n : s_n\}$, the state-space will represent the amount of recurrences in the sequence $\mathcal{K} : \{S_1 : k = 1, S_2 : k = 2, .., S_n : k = n\}$.

Regardless of what model will implement it, by using this definition and combining it with Bayes' theorem (which must hold for any segment of probability theory) the following holds:

$$P(K = k+1|K \geq k) = \frac{P(K = k|K = k+1) \cdot P(K = k+1)}{P(K \geq k)} \tag{4}$$

If $k+1$ recurrences have already been observed, them $P(K = k|K = k+1)$ must be 1 and the term dissapears. This means that equation 4 will always equal:

$$P(K = k+1|K \geq k) = \frac{P(K = k+1)}{P(K \geq k)} \rightarrow P(K = k+1) = P(K \geq k) \cdot P(K = k+1|K \geq k) \tag{5}$$

This makes sense, to get to $k+1$ recurrences, you first need to get to k recurrences and then from k to $k+1$ recurrences. Due to the interest in modelling the recurrence probabilities, the idea is proposed that a density function should be used to model this.

A simple recurrence probability model should suffice the requirements set by general probability theory. This means that also amounts of recurrences can be computed by using equation 6:

$$P(K = k) = P(K = 1) \cdot P(K = 2|K \geq 1) \cdot \ldots \cdot P(K = k|K \geq k-1) \tag{6}$$

2.6 Fitting distributions to recurrence probabilities

The second part of the hypothesis is that the recurrence probabilities not only change based on the amount of previous recurrences, but actually follow a distribution. More specifically, the hypothesis that recurrence probabilities will be modelled using functions $f(x)$ as displayed in 7.

$$P(K = k+1|K \geq k) = f(k) + \varepsilon \tag{7}$$

Equation 7 can be used to estimate probabilities for specific recurrence quantities, e.g.:

$$P(K = k) = P(K = 1) \cdot \prod_{i=1}^{k-1} f(i) \tag{8}$$

This means that a (non-linear) regression will be preformed onto the probabilities of the Markov chain.

Not every density functions can be used, as certain characteristics are required. Below is a list for what minimally is required:

1. $f(k)$ is well defined $\forall k \in 1, 2, \ldots$.
2. $0 < f(k) < 1, \forall k \in 1, 2, \ldots$ because it represents probabilities

3. $f(k)$ can approach 0 or 1, but only if it does so as a limit: $\lim_{k\to\infty} f(x) = 1$ or $\lim_{k\to\infty} f(x) = 0$. If $\lim_{k\to\infty} f(x) = 1$, then $P(K \leq k) = \sum_{k=2}^{\infty} \prod_{i=1}^{k} f(k)$ must converge with a limit $L \leq 1$.

A bigger explanation and proof for these points can be found in Appendix 1. Note that it is not required that the density function will start at a specific value. With these requirements in place, density functions can be formulated, which are appropriate for fitting recurrence probabilities.

3 Subproblem 2a: Exploring recurrence probability Markov chains using data

3.1 Method: Exploring augmenting classical Markov chains using data

Now that properties of an ideal model have been specified, possible models will be selected for exploration. This will happen as a proof of concept using a selection of data. The data that has been selected is the "Hospital Billing - Event Log" from the 4TU repository. The data contains 100.000 event sequences with a total of 415.359 event occurences. This dataset has been selected for its variety in possible events (18 distinct events) and its average and maximum length of event sequences (217) and the presence of recurring behaviour.

As with any data science project, models will require verification. Therefore, this report will also have literature that provides quantitative measures on how proposed models could be verified. Possible models will be tested on their possibility, along with an exploratory attempt to verify them through literature-based proposed quantitative measures.

3.2 From classical Markov chains to recurrence probability Markov chains

The first-order Markov model has a matrix, which describes the probabilities to move from state s_i to state s_{i+1}. However, modelling the probability of having another recurrence is the actual goal. Earlier a description was provided on how this issue could be resolved by changing the view on what the state-space is in a process model. Therefore, the proposal is to explore a default first-order Markov chain with a different view on the event-space: the recurrence probability Markov chain (RPMC). Within this new Markov chain, $P(K = k+1|K \geq k)$ can be directly observed. Here the complementary rule can be used to compute $1 - P(K = k+1|K \geq k) = P(\text{no more loops})$ where the stop-state (when required for insights) will be indicated in Markov chains as "stop".

3.3 Recurrence probability Markov chains: Estimation

The probability matrix of Markov chains is computed using the Maximum Likelihood Estimator (MLE) techniques (Teodorescu, 2009). This MLE technique is also

used for higher-order Markov chains although using higher-order Markov chains introduces extra in-model parameters. This often leads to a balance between increase of performance due to increased memory and decrease of performance due to an overload memory. However, because MLE-techniques apply, the recurrence probability Markov chain will have proper estimators as long as it is fed the right data.

To make sure the RMPC is provided with the right data, a transformation of the state-space is required. This means that an event sequence will no longer be represented by $\mathcal{S} : \{S_1 : A, S_2 : B, .., S_n : s_n\}$. Instead, event sequences will be reprented by ($\mathcal{K} : \{S_1 : k = 1, S_2 : k = 2, .., S_n : k = n\}$. Practically, this would transform event sequences, for example $\langle A, B, C, D, C, E \rangle$, into $\langle C, C, stop \rangle$ when specifically looking at recurrences for event C.

3.4 Recurrence probability Markov chains: Curve fitting

Given the fitted Markov chain, a density function should be fitted by the following recurrence probabilities:

$$\begin{aligned} &P(2|1) = \hat{p}_{21} \\ &P(3|2) = \hat{p}_{32} \\ &\dots \\ &P(K = k+1|K \geq k) = \hat{p}_{k+1,k} \end{aligned} \tag{9}$$

In equation 9, $\hat{p}_{21}$ actually represents the amount of event sequences that had another recurrence divided by the amount of event sequences and thus: $\hat{p}_{21} = \frac{x}{n}$. $\hat{p}_{k+1,k}$ is modelled as if it were to lie on a (non-linear) line. Like any other model, it needs to be verified through some goodness of fit statistics (e.g. R^2 for GLM).

Because probability densities will be modelled as a non-linear curve using regression, the default Mean Squared Error (MSE) statistic will be used, expressed with the following equations:

$$P(K = k+1|K \geq k) = f(k) + \varepsilon \tag{10}$$

$$\hat{\varepsilon} = \hat{p}_{k+1,k} - f(k) \tag{11}$$

$$\hat{\varepsilon}^2 = (\hat{p}_{k+1,k} - f(k))^2 \tag{12}$$

$$MSE = \frac{\sum_{i=1}^{n} \hat{\varepsilon}^2}{n} \tag{13}$$

Previously, the density function $f(k)$ has been written down using a slight abuse of notation, as it is expected to be a parameter-dependent function. The parameter-set of any density function $f(k)$ will be denoted as $\bar{\lambda} = \{\lambda_1, \lambda_2, \dots, \lambda_j\}$.

Having done this, parameter estimation methods such as MLE or numerical methods can be applied and hence, can be used to determine estimators for the density functions.

3.5 Recurrence probability Markov chains: Results

The Markov matrix in table 2 is a result of modelling data into the classic first-order Markov chain.

to \ from	BILLED	CHANGE DIAGN	CHANGE END	CODE ERROR	CODE NOK	CODE OK	DELETE	EMPTY	FIN	JOIN-PAT	MANUAL	NEW	REJECT	RELEASE	REOPEN	SET STATUS	STORNO	ZDBC_BEHAN
BILLED	0.02	-	-	0.33	0.35	0.95	-	-	-	0.03	0.18	-	0.40	0.01	-	-	0.31	1.0
CHANGE DIAGN	-	0.06	0.03	-	-	-	-	-	-	0.02	-	0.54	-	-	0.05	-	-	-
CHANGE END	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.03	-	-
CODE ERROR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CODE NOK	-	-	-	-	0.10	-	-	-	-	-	0.01	-	-	0.05	-	0.07	-	-
CODE OK	0.22	-	-	0.01	0.13	-	-	-	-	-	0.20	-	-	0.94	-	0.02	-	-
DELETE	-	0.04	0.26	-	0.03	-	0.4	1.0	-	0.01	0.01	0.06	-	-	0.24	0.06	-	-
EMPTY	-	-	-	-	0.12	-	-	-	-	-	-	-	-	-	-	-	-	-
FIN	-	0.90	0.71	0.01	-	-	-	-	-	0.09	0.02	0.38	-	-	0.70	0.08	-	-
JOIN-PAT	-	-	-	-	-	-	0.6	-	-	0.85	-	-	-	-	-	-	-	-
MANUAL	0.02	-	-	0.03	-	-	-	-	-	-	0.25	-	0.01	-	-	-	-	-
NEW	-	-	-	-	-	-	-	-	-	-	-	0.02	-	-	-	-	-	-
REJECT	-	-	-	-	-	-	-	-	-	0.01	-	-	0.01	-	-	-	0.67	-
RELEASE	-	-	-	0.03	-	-	-	-	0.99	-	0.29	-	0.13	-	-	0.18	-	-
REOPEN	-	-	-	0.59	0.17	0.04	-	-	0.01	-	0.02	-	0.41	0.01	-	0.10	-	-
SET STATUS	-	-	-	-	0.10	-	-	-	-	-	-	-	0.03	-	-	0.47	0.01	-
STORNO	0.74	-	-	-	-	-	-	-	-	-	0.01	-	-	-	-	-	0.01	-
ZDBC_BEHAN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 2: First-order Markov chain for the event sequences in the hospital billing dataset.

To get as much insights into the dataset as possible, process discovery algorithms have also been used. When using the Alpha Miner algorithm similarly to how it was previously used on the Loan application dataset, the algorithm does not provide us with an actual ”insightful” model as can be seen in figure 3.

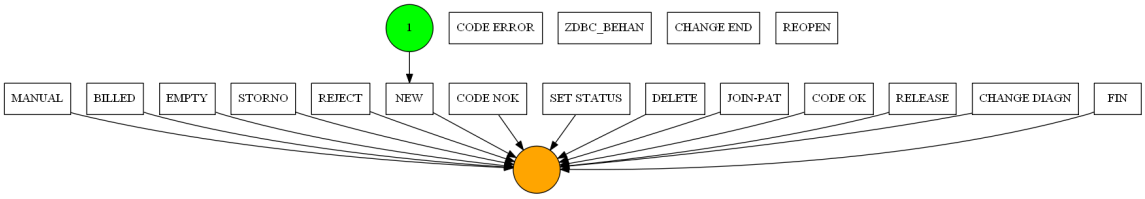

Figure 3: Discovered Petri net from Hospital Billing data using the Alpha Miner.

After several attempts with a variety of algorithms, only the Heuristics Miner provides an insightful visualization.

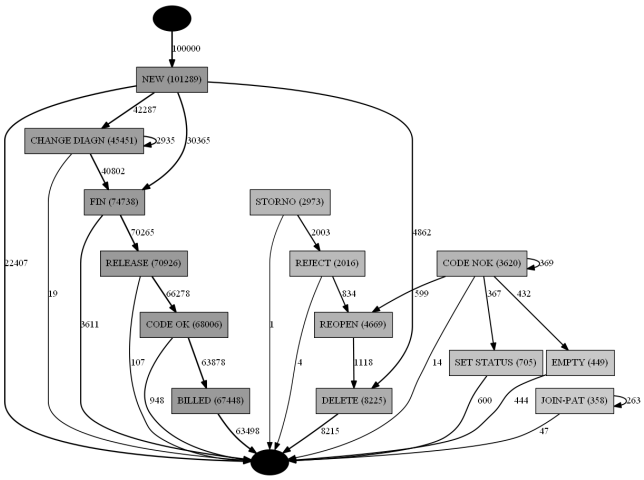

Figure 4: Discovered Process model from Hospital Billing data using the Heuristics Miner.

Unfortunately, it can be observed that the Heuristics miner is missing a lot of recurrences in the data. This is most likely due to the hyperparameter selected for the algorithm. For example, take the frequency table of recurrences, given in table 3.

	1	2	3	4	5	6	7	8	9	10	15	16	17	18	19
BILLED	62657	1870	278	38	5	4	-	2	-	-	-	-	-	-	-
CHANGE DIAGN	39443	2632	214	23	2	-	-	-	-	-	-	-	-	-	-
CHANGE END	30	1	2	-	-	-	-	-	-	-	-	-	-	-	-
CODE ERROR	62	3	1	1	-	-	-	-	-	-	-	-	-	-	-
CODE NOK	2643	316	70	8	10	2	2	1	1	1	-	-	-	-	-
CODE OK	60167	3229	341	62	16	5	-	-	-	-	-	-	-	-	-
DELETE	8217	4	-	-	-	-	-	-	-	-	-	-	-	-	-
EMPTY	449	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FIN	67816	2775	326	65	18	2	1	2	1	-	-	-	-	-	-
JOIN-PAT	6	-	4	85	-	-	-	-	-	-	-	-	-	-	-
MANUAL	166	53	11	5	3	3	2	-	-	-	-	-	-	-	-
NEW	99806	134	17	14	4	2	2	1	1	1	1	1	1	2	1
REJECT	1689	114	19	2	3	2	1	-	-	-	-	-	-	-	-
RELEASE	63766	2936	314	59	13	5	1	1	-	-	-	-	-	-	-
REOPEN	3626	372	59	20	4	-	2	1	-	-	-	-	-	-	-
SET STATUS	579	33	16	3	-	-	-	-	-	-	-	-	-	-	-
STORNO	2193	306	38	5	3	2	1	-	-	-	-	-	-	-	-
ZDBC_BEHAN	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 3: Frequency table for how often a number of recurrences appeared.

Table 3 shows for every event how many event sequences there are with k recurrences of the event. Take the event "BILLED", 62.657 event sequences have occured with only a single "BILLED" event, 1870 with two "BILLED" events, and so on. This means that this recurrence behaviour is poorly represented within the process discovery algorithms. A second point of attention is that the columns are jumping from 10 to 15. This is not an error. It means that for the event "CODE NOK" there is 1 event sequence that contained "CODE NOK" 10 times and 1 that contained it 15 times. However, no event sequences contained "CODE NOK" 11, 12, 13 or 14 times. This means that with this data and this model, $P(K = 11|10) = 0.5$ and $P(K = 12|11) = P(K = 13|12) = P(K = 14|13) = P(K = 15|14) = 1$. There are other events that do not make sense, there are multiple event sequences which contain 100+ recurrences of "NEW" only to end with a frequent normal ending ('FIN', 'RELEASE', 'CODE OK', 'BILLED').

The "BILLED" event appears to behave like a normal event and will therefore be the event of focus. The recurrence probability Markov chain given in table 4. Because visual results are valued, the recurring probabilities have been plotted in figure 5.

	1	2	3	4	5	6	7	8
2	0.033876	0.000000	0.000000	0.00000	0.000000	0.000000	0.0	0.0
3	0.000000	0.148839	0.000000	0.00000	0.000000	0.000000	0.0	0.0
4	0.000000	0.000000	0.149847	0.00000	0.000000	0.000000	0.0	0.0
5	0.000000	0.000000	0.000000	0.22449	0.000000	0.000000	0.0	0.0
6	0.000000	0.000000	0.000000	0.00000	0.545455	0.000000	0.0	0.0
7	0.000000	0.000000	0.000000	0.00000	0.000000	0.333333	0.0	0.0
8	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	1.0	0.0
stop	0.966124	0.851161	0.850153	0.77551	0.454545	0.666667	0.0	1.0

Table 4: Markov matrix for recurring "BILLED" in the Hospital billing dataset.

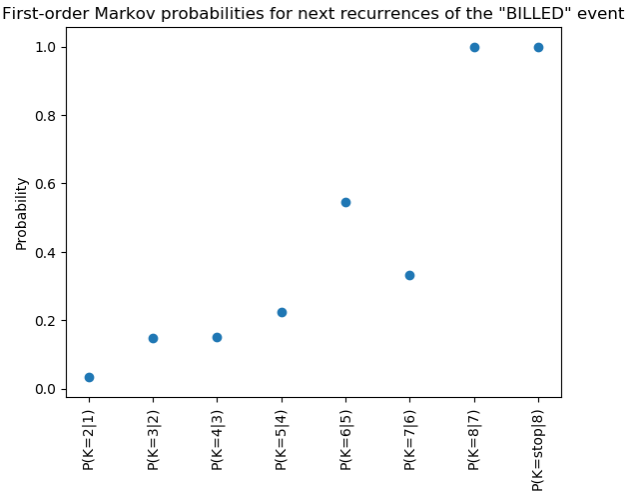

Figure 5: A visualization of the probabilities for ”BILLED” recurrences.

From the overall frequency table, it is noticed that the likelyhood of ”BILLED” recurring a large amount (5+) is rare and every larger recurrence is in general even rarer. However, what is noticed from its Markov chain and visualization is that the probability for another recurrence goes up as the number of recurrences goes up. This means that once you get to seeing 3 recurrences, which is uncommon by itself $P(K = 3) = 0.005$, it is common to see another recurrence from $k = 3$ and another from $k = 4$ and so forth.

As is seen, recurrence probabilities can be modelled for a single event (”BILLED”). However, process models are often complex with many events. Therefore, it is important to view recurrence probabilities from the perspective of multiple events. To visualize this, two plots were developed. Recurrences probabilities were taken from the Markov matrices similar to 4 and placed them in a single plot. Some of these lines will fall ”short” as some events simply recur longer than others as can be seen in 3 (e.g. ”BILLED” has a maximum of 4 recurrences and ”CODE ERROR” a maximum of 4). Important to note is that certain events ”skip” states, e.g. ”BILLED” going only having $k = 6$ and $k = 8$ but no $k = 7$. Therefore, 2 plots were made: one with interim and exit probabilities of 1 included, and one where probabilities of 1 were replaced by NaNs. All of this is shown in figure 6 and figure 7, respectively.

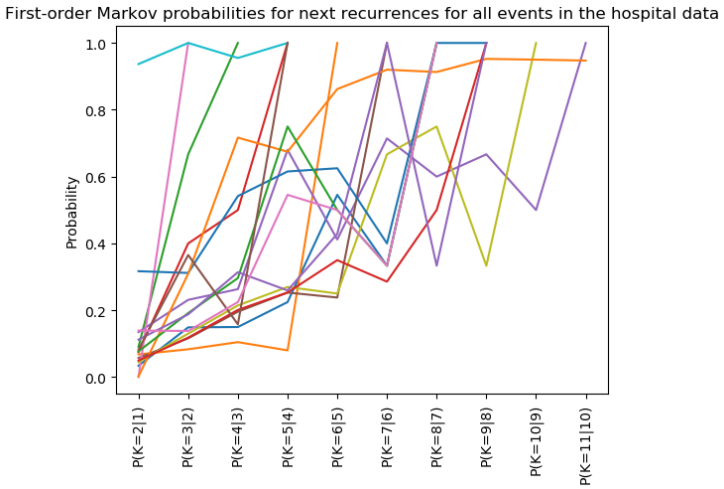

Figure 6: Probabilities for next another recurrence, interim and exit probabilities of 1 included.

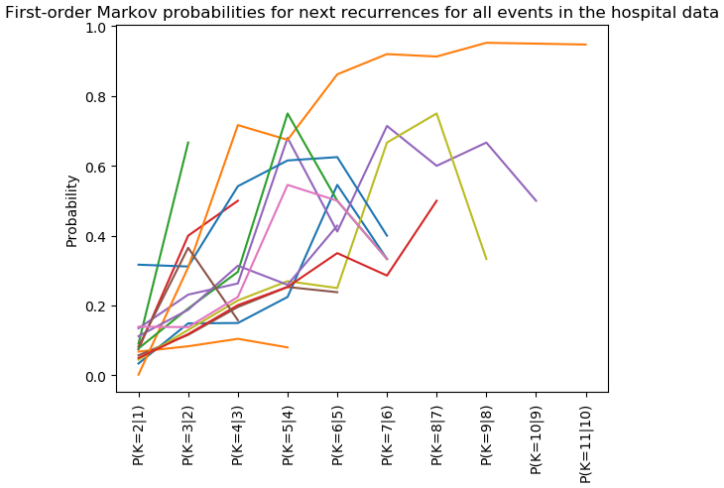

Figure 7: Probabilities for next another recurrence, interim and exit probabilities of 1 excluded.

It is observed that for the hospital dataset, the probability of seeing another recurrence goes up based on how many recurrences it has already seen. This shows that using a classical Markov chain approach (e.g. seeing the events as a random walk) as was done in table 2 is an unsuitable approach for this dataset. It also shows that the proposed definition of the reccurence probability Markov chain is a valueble addition to the classical view of process models, albeit for specific process characteristics.

One of the proposed questions was whether recurrence probabilities can be modelled as a distribution. The event ”CODE NOK” provides the most data points (10), has no discontinuities and roughly indicates normal behaviour as can be seen in figure 8.

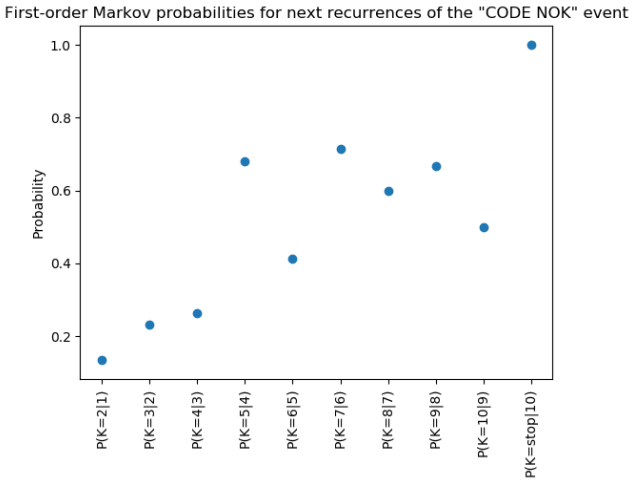

Figure 8: A visualization of the probabilities for ”CODE NOK” recurrences.

The points seem to indicate a low recurrence probability for low amounts of recurrences and an increasing recurrence probability as the amount of recurrences increases.

As an exploratory suggestion, the selected density function is $f(k) = 1 - e^{-\lambda \cdot k}$. It is a function that approaches zero when k goes to zero, approaches 1 as k increases and is monotonically increasing. Function optimization will look for an optimal value λ that fits the points the best. This resulted in $\hat{\lambda} = 0.15$, $Var(\hat{\lambda}) = 0.0005$ and MSE of 0.02. The resulting optimal density line is visualized in figure 9

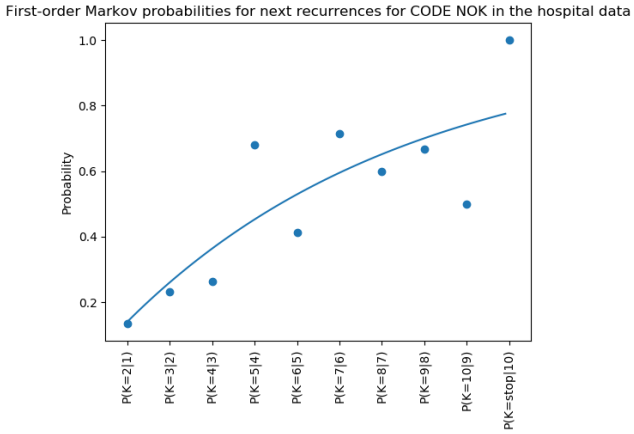

Figure 9: A visualization of the probabilities for ”CODE NOK” recurrences and the fitted optimal density line.

With this determined density function, recurrence probabilities can be calculated for recurrences beyond the maximum number of recurrences in the data. For example, ”CODE NOK” has a maximum of 10 recurrences. Using equation 7 and equation 8, quantitative estimates can be provided for onward recurrence probabilities (e.g. $P(K = 11|10) = 0.778$ as well as onwards probabilities for the number of recurrences in a single event sequence (e.g. $P(K = 11) = 0.00049$).

4 Subproblem 2b: Exploring higher-order Markov chains using data

4.1 Method: Exploring high order Markov chains using data

In the results of subproblem 2a, recurrence probabilities were modelled for the event ”BILLED” from the hospital data, using only a first-order Markov chain. In this chapter, an alternative model (the higher-order Markov chain) will be explored for suitability for modelling recurrence probabilities. As could be seen earlier, the higher-order Markov chain uses more ”memory” for its probability estimations.

4.2 From higher-order Markov chains to recurrence probability Markov chains

As specified earlier, the Markov property defines the probability of the next state based on the current state. Increasing the order of a Markov chain means incoorporating more memory. Recall that the Markov property for a second-order Markov chain was defined as equation 14.

$$P(X_{n+1} = x|X_1 = x_1,..,X_n = x_n) = P(X_{n+1} = x|X_n = x_n, X_{n-1} = x_{n-1}) \qquad (14)$$

Using this definition into the state-space transformation would give the recurrence probability definition in equation 15.

$$P(K = k+1 | K \geq k, K \geq k-1) \tag{15}$$

If $K \geq k$ is given then $K \geq k-1$ does not provide any new information. This means higher-order Markov chains cannot be used directly for recurrence modelling, but only as an increased memory Markov chain. However, it may still be worthwhile investigating how recurrence probabilities could be retrieved, albeit indirectly and possibly less than ideal.

For example, take a second-order Markov chain and say that the objective is to obtain the probability of another recurrence of event A (in a possible state-space A to G), after it visited another state first. Then the probability becomes as in equation 16:

$$P(X_{n+1} = A | X_n \neq A, X_{n-1} = A) = \sum_{i=B}^{G} P(X_{n+1} = A | X_n = i, X_{n-1} = A) \tag{16}$$

This definition can be extended if the order is increased. For example, what is the probability of another recurrence of event A (in a possible state-space A to G), after it visited another state or two other states first?

$$P(X_{n+1} = A | X_n \neq A, X_{n-1} \neq A, X_{n-2} = A) = \sum_{i=B}^{G} \sum_{j=B}^{G} P(X_{n+1} = A | X_n = i, X_{n-1} = j, X_{n-2} = A \tag{17}$$

Admittedly, it is less than ideal but can still provide insights when looking for specific recurrences. Therefore, this model will be explored, albeit in less detail than the recurrence probability Markov chain as it is deemed useful when wanting to explore specific recurrences.

4.3 Higher-order Markov chains: Verification statistics

(Teodorescu, 2009) showed that higher-order Markov chains are estimated using MLE-techniques. This means that a verification is required for why a certain order is deemed better than another. A display on what methods could be used for estimating the "optimal order" for a Markov chains are in (Singer, Helic, Taraghi, & Strohmaier, 2014). They specified techniques that essentially come down to 1v1-comparisons model: an l-order vs. m-order comparison. (Singer, 2013) stated that two common comparison statitics that use the LLR are the Akaike's Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Both of these information criterions make use of the LLR statistic.

This means that if an initial exploratory data analysis shows that the modelling makes sense and produces sensible results, these statistics will be used to verify model comparisons. Note that this will be performance comparisons between models and not neccesarily between model performances.

4.4 Higher-order Markov chains: Results

Before getting deeper into the higher-order Markov chains, it is important to note that fully visualizing the Markov chain probability matrix for higher-orders is unfeasible. The amount of possible combinations scales exponentially based on the number of distinct events. Even for one of the smaller datasets such as the "Loan Application" data, the amount of entries scales with a term of 7^m for a Markov chain of order m. Medium sized datasets such as "Hospital Billing" scales in terms of 18^m. Even removing zero-possibility states (that neither have incoming nor outgoing probabilities) does not provide a guaranteed decent manner of displaying these results. Therefore, only highlights or the lack of highlights will be shown in the results.

The training of models is started from order 1 to 8, as those are maximum number of recurrences that show up in the training data. As specified earlier, a l-order vs. m-order comparison is used by computing the AIC and BIC from the loglikelihood ratio. A first glance of the results are shown in table 5, where the results are sorted in an ascending order on AIC.

	l	**m**	**lnm**	**AIC**	**BIC**
13	2	8	148321.081057	-3.746785e+11	-2.156824e+12
20	3	8	102615.580594	-3.746784e+11	-2.156822e+12
6	1	8	302137.161984	-3.746784e+11	-2.156823e+12
27	4	8	65926.689004	-3.746750e+11	-2.156803e+12
34	5	8	34239.267944	-3.746144e+11	-2.156454e+12
41	6	8	12930.637948	-3.735222e+11	-2.150167e+12
48	7	8	5297.593367	-3.538632e+11	-2.037000e+12
12	2	7	143023.487690	-2.081533e+10	-1.198233e+11
19	3	7	97317.987227	-2.081519e+10	-1.198223e+11
5	1	7	296839.568617	-2.081518e+10	-1.198232e+11

Table 5: First results of l-order vs m-order comparison.

If the AIC score is low, then the l-order model outperforms the m-order model (Singer, 2013). Knowing this, it can be immedietly deduced that 8-order models perform bad, as every comparison (order 1 to 7 vs. order 8) is present in this top. Using this same logic, the higher-orders (7 to 4) could be removed, and the rest of the results are shown in table 6. Note that using BIC (for now) leads to the same conclusion.

	l	m	lnm	AIC	BIC
2	2	1	-153816.080927	-143412.080927	-9.392584e+04
3	2	3	45705.500463	-141566.499537	-1.032319e+06
4	3	1	-199521.581391	-1845.581391	9.383929e+05
1	1	3	199521.581391	1845.581391	-9.383929e+05
5	3	2	-45705.500463	141566.499537	1.032319e+06
0	1	2	153816.080927	143412.080927	9.392584e+04

Table 6: Filtered results of l-order vs. m-order comparison.

As can be seen in table 6, the AIC concludes that the second-order model outperforms both the first-order Markov chain and third-order Markov chain.

	l	m	lnm	AIC	BIC
3	2	3	45705.500463	-141566.499537	-1.032319e+06
1	1	3	199521.581391	1845.581391	-9.383929e+05
2	2	1	-153816.080927	-143412.080927	-9.392584e+04
0	1	2	153816.080927	143412.080927	9.392584e+04
4	3	1	-199521.581391	-1845.581391	9.383929e+05
5	3	2	-45705.500463	141566.499537	1.032319e+06

Table 7: Filtered results of l-order vs. m-order comparison.

Table 7 provides the last scores but now sorted on BIC. Here it can be concluded that both the first-order and second-order outperform the third-order model. After that, the second-order Markov chain outperforms the first-order. Therefore, it can be concluded that to predict recurrences based on the Hospital dataset, a second-order Markov chain is our best pick.

5 Discussion

The goal of this project was in two-fold: to show that current mainstream process models do not account for specific behaviour, and to provide a proof of concept on how to model this behaviour.

Using literature and a exploratory data analysis, it was possible to show that current mainstream process models do not account enough for shifting recurrence probabilities. That is, the probability for seeing an event $k+1$ times given that it has been observed k times changes with k. From the literature, it showed that process discovery algorithms mainly focus on a variety of quantitative measures that provide insights into general model performance. Using an example, it was possible to show that these measures fail to account for very distinct characteristics that can be expected from process models.

Following this, the required properties of an ideal model needed to describe shifting recurrence probabilities were explained and defined mathematically. This ideal model

description can and should be used alongside current mainstream process models and not as a standalone model. It provides a unique view into a specific process model characteristic (recurrent behaviour), but only in a limited manner.

Using existing models such as classic first-order Markov chains and higher-order Markov chains, it was possible to provide proofs of concept for modelling recurrent behaviour. As a proof of concept, only recurrences of length 1 were considered. This means that the scope of this report solely covers the recurrences of single events. Hence, results from this report cannot trivially be generalized to recurrences of multiple events. However, a big opportunity lies within the combining of the results from this report with the work done by (Ferreira et al., 2012) and (Ferreira et al., 2014). Such a combination could yield recurrences of nested Markov models, rather than singular events.

Using a transformation of the state-space, it was possible to change classic first-order Markov chains from describing event sequences to describing recurrences. For these results, MLE-techniques were applicable. Unfortunately, little to no literature deals with certainty for the probability-estimators of a Markov chain. Therefore, this report cannot provide confidence intervals nor can it say anything about whether probabilities converge (it could be possibly asymptotically). This means that it is not possible to impose error bounds and indicate how these scale as the sample size grows. It is therefore important to use the results from this paper only on event logs that have a sufficient number of event sequences in the event log. Unfortunately, it is also not possible to indicate what a "sufficient" number would be or how this scales for example on number of distinct events.

After having a recurrence probability Markov chain, the recurrence probabilities could be used to fit a density function. This could be done by using non-linear regression and basic verification statistics such as the Mean Squared Error. Given certain required properties, these density functions allowed us to generalize recurrences. It can be used to quantitatively estimating how rare unobserved event sequences can be. Here lies the opportunity for developing the application of techniques such as cross-validation. For example, training recurrence probability Markov chains on data-splits and comparing the resulting fitted density functions. Unforunately, the rarity estimation technique currently only generalizes for single events. Hence, it poses a challenge to develop a technique which incorporates multiple density functions or incorporates a joint density functions, such that multiple distinct events can be used for rarity estimation.

Last, higher-order Markov chains were used to see the effects of increasing memory. This was done with the knowledge that the model did not fit the ideal model properties, but was continued as it was deemed useful when wanting to look at specific recurrent behaviour. Using literature, it was possible to find an verifiable optimal order for a dataset. However, in the beginning of the report it was proven that when event logs contain parallel behaviour, Markov chains tend to allow for behaviour that is not encapsulated by the event-log. While modelling behaviour that is not encapsulated by the log was the goal for the recurrence probability Markov chain, this form of generalization can be seen as multidimensional. Hence, when using higher-order Markov chains to describe event-log based process models, it must be noted that in-depth verification of precision is a strong requirement.

6 Conclusion

Many real life processes contain recurrences of events. The probability of having another recurrence in an event sequences can change, based on how many recurrences have already occured in an event sequence. This is demonstratable using the Hospital Billing dataset, which shows that recurrence probabilities change based on how many prior recurrences occured.

In this paper, arguments is provided for how current mainstream process models do not account for these shifting recurrence probabilities. Many process discovery algorithms focus on general model performance measures, which do not necessarily reflect process characteristics such as shifting recurrence probabilities.

A mathematical description is given of the requirements for an ideal model to describe shifting recurrence probabilities. This description is then used as argument to transform the event-space of a classic first-order Markov chain to fit a recurrence probability Markov chain. This is demonstrated with an illustrative example using event-log data, as proof of concept. This paper also provides a basis on how this model can be used to generalize recurrence behaviour.

Besides defining and exploring a proposed ideal model, there is an exploration of higher-order Markov chains, how these can be used for recurrences and how to determine the optimal order of a higher-order Markov chain.

The paper ends with a discussion, highlighting the limits of recurrence probability Markov chains and higher-order Markov chains. There is also a suggestion on which and how already existing literature can be combined for new closely related extensions.

References

Ayo, F. E., Folorunso, O., & Ibharalu, F. T. (2017). A probabilistic approach to event log completeness. *Expert Systems with Applications*, *80*, 263 - 272. Retrieved from http://www.sciencedirect.com/science/article/pii/S0957417417301938 doi: https://doi.org/10.1016/j.eswa.2017.03.039

Bazzi, M., Blasques, F., Koopman, S. J., & Lucas, A. (2017). Time-varying transition probabilities for markov regime switching models. *Journal of Time Series Analysis*, *38*(3), 458-478. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/jtsa.12211 doi: 10.1111/jtsa.12211

Buijs, J. C. A. M., van Dongen, B. F., & van der Aalst, W. M. P. (2014). Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity. *International Journal of Cooperative Information Systems*, *23*(01), 1440001. Retrieved from https://doi.org/10.1142/S0218843014400012 doi: 10.1142/S0218843014400012

Conforti, R., Rosa, M. L., & t. Hofstede, A. H. M. (2017, Feb). Filtering out infrequent behavior from business process event logs. *IEEE Transactions on Knowledge and Data Engineering*, *29*(2), 300-314. doi: 10.1109/TKDE.2016.2614680

Ferreira, D. R., Szimanski, F., & Ralha, C. G. (2012). A hierarchical markov model to understand the behaviour of agents in business processes. In *Business process management workshops.*

Ferreira, D. R., Szimanski, F., & Ralha, C. G. (2014, Oct 01). Improving process models by mining mappings of low-level events to high-level activities. *Journal of Intelligent Information Systems*, *43*(2), 379–407. Retrieved from https://doi.org/10.1007/s10844-014-0327-2 doi: 10.1007/s10844-014-0327-2

Rabiner, L. R. (1989, Feb). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257-286. doi: 10.1109/5.18626

Raftery, A. E. (1985). A model for high-order markov chains. *Journal of the Royal Statistical Society. Series B (Methodological)*, *47*(3), 528–539. Retrieved from http://www.jstor.org/stable/2345788

Singer, P. (2013). Order estimation for markov chain models. Retrieved from https://medium.com/@ph_singer/order-estimation-for-markov-chain-models-6cde3ad2410b

Singer, P., Helic, D., Taraghi, B., & Strohmaier, M. (2014, 07). Detecting memory and structure in human navigation patterns using markov chain models of varying order. *PLOS ONE*, *9*(7), 1-21. Retrieved from https://doi.org/10.1371/journal.pone.0102070 doi: 10.1371/journal.pone.0102070

Teodorescu, I. (2009, 05). Maximum likelihood estimation for markov chains.

vanden Broucke, S. K. L. M., De Weerdt, J., Vanthienen, J., & Baesens, B. (2013, April). A comprehensive benchmarking framework (cobefra) for conformance analysis between procedural process models and event logs in prom. In *2013 ieee symposium on computational intelligence and data mining (cidm)* (p. 254-261). doi: 10.1109/CIDM.2013.6597244

van der Aalst, W. (2016). *Process mining: Data science in action* (2nd ed.). Springer Publishing Company, Incorporated.

7 Appendix

7.1 Appendix 1

For fitting distributions to recurrence probabilities, 3 specific requirements were posed. $f(k)$ must be well defined, must have a domain or approach the domain bounds as a limit. In this part of the appendix the specific focus will lie on property 2 and 3.

This is because of the meaning of equation 8. The sum of this term must converge, hence:

$$P(K \leq k) = \sum_{k=2}^{\infty} P(K = k) \leq 1 \tag{18}$$

7.1.1 Property 2

From equation 5, if $\lim_{k\to\infty} f(x) \neq \{0,1\}$, then the ratio test will provide the following:

$$\lim_{k\to\infty} \frac{\prod_{i=1}^{k} f(k)}{\prod_{i=1}^{k-1} f(k)} = \lim_{k\to\infty} f(k) \tag{19}$$

Hence, if $\lim_{k\to\infty} f(k) \neq \{0,1\}$, then $P(K \leq k)$ converges.

7.1.2 Property 3

An argument similar to that of property 2 can be used when $\lim_{k\to\infty} f(k) = 0$, then the ratio-test concludes convergence.

However, if $\lim_{k\to\infty} f(k) = 1$ then the ratio-test is inconclusive. To see whether this holds, a substitute to the ratio-test is needed. Otherwise, if series that do prove to converge can often be fixed by adding an "intercept"-parameter.

Take for example $f(k) = 1 - e^{-\lambda \cdot k}$. The ratio test is inconclusive. Using the Raabe-Duhamel's test the following is obtained:

$$b_k = k \cdot (\frac{a_k}{a_{k+1}} - 1) \tag{20}$$

Taking $a_k = \prod_{i=1}^{k} f(k)$, $\lim_{k\to\infty} b_k = 0$ and hence the series diverges. However, for density functions like $f(k) = 1 - e^{-\lambda \cdot k}$, this can be fixed by not letting it not converge to 1. This can be done by adding a finite negative term. e.g. $f(k) = 1 - e^{-\lambda \cdot k} - 0.000001$. Doing so will have no significant impact on the results. It is an increasing function approaching 1 and hence, it's effects are negligible. With this correction term, $\lim_{k\to\infty} b_k = \infty$ and hence the series converges.

Concluding density functions must be meet the requirement of cumulative probability function convergence ($\lim_{k\to\infty} P(K \leq k) \leq 1$).