Finding Markovian Models for Insurance Processes by Expanding State Spaces

Von der Fakultät für Mathematik und Naturwissenschaften
der Carl von Ossietzky Universität Oldenburg
zur Erlangung des Grades und Titels eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

angenommene Dissertation

von Herrn Marius Pluhar
geboren am 03.07.1993 in Bremen.

Betreuender Gutachter: Prof. Dr. Marcus C. Christiansen
Zweitgutachter: Prof. Dr. Peter Ruckdeschel
Tag der Disputation: 01.10.2020

"Make no little plans; they have no magic to stir men's blood and probably themselves will not be realized. Make big plans; aim high in hope and work, remembering that a noble, logical diagram once recorded will never die, but long after we are gone be a living thing, asserting itself with ever-growing insistency. Remember that our sons and our grandsons are going to do things that would stagger us. Let your watchword be order and your beacon beauty."

— Daniel H. Burnham

Acknowledgements

Pursuing my doctoral degree has been a life-changing experience for me. Since the start of my PhD studies I have not only been able to deepen my existing mathematical knowledge and to obtain new mathematical insights, but constantly pushing myself to new limits has also made me grow as a person. Without the support I received from many people and parties this would not have been possible.

First I would like to express my very great appreciation to the SIGNAL IDUNA insurance. They generously provided the data set that enabled me to illustrate the theoretical results of my research. Without their contribution this thesis could not have been completed. In particular I want to thank Dr. Christian Scholz and Dr. Karsten Dietrich. Their thoughtful questions and remarks have led to enhancements of this work.

Next I am very grateful for the continuous assistance and the helpful feedback given by my main supervisor Professor Marcus C. Christiansen and my co-supervisor Professor Peter Ruckdeschel. Both of them supported me and my research in an exemplary manner. They contributed countless ideas that clearly increased the quality of my thesis and our talks enriched my work.

In addition to my supervisors, my appreciation also extends to the participants of the "Joint PhD Seminar in Statistics, Actuarial and Financial Mathematics", the "Joint PhD Seminar in Statistics and Stochastics" and the "Oldenburger Nachwuchsworkshop für Versicherungs- und Finanzmathematik" for helpful discussions and suggestions.

Furthermore I would like to thank my fellow PhD students and friends for their stimulating remarks, their steady encouragement and the fun we had together in the last years.

Last but not least, I feel immense gratitude towards my parents and my brother as they have always unconditionally supported me in every possible way.

Abstract

(English)

In insurance mathematics time-discrete stochastic processes are often assumed to be first-order Markovian. While the actual data does most likely not harmonize with the Markov assumption, ignoring this yields models that are easy to handle and easy to communicate to customers. By switching over to Markov processes of higher order this discrepancy can be reduced, but the complexity of higher-order models increases exponentially and the major advantages of a first-order model are lost.
Variable Length Markov Chains (VLMC) are a subclass of Markov processes. They are able to fully display dependencies in time-homogeneous data while being less complex than a Markov process of sufficiently large order. But VLMC fail to meet requirements of insurance applications, e.g. they cannot display time-dependencies in a natural way.
In order to meet those requirements, we extend the concept of VLMC and introduce a new class of models: time-inhomogeneous Variable Length Markov Chains (tiVLMC). We propose and implement a fitting algorithm for tiVLMC within an uncensored and a censored data setting. The calculation of prospective reserves is discussed and we train and tune tiVLMC-models on two real life data sets. The first data set documents long-term care insurances and the second data set documents joint life insurances. We also develop smoothing procedures for tiVLMC-models that ensure their interpretability and make them easy to communicate to potential customers.

Abstract

(German)

Zeitdiskrete stochastische Prozesse werden in der Versicherungsmathematik oft mittels Markov-Prozessen erster Ordnung modelliert. Obwohl die Datenlage und die Markov-Annahme in den meisten Fällen nicht zusammenpassen, können so schlanke und einfach kommunizierbare Modelle gewonnen werden. Die Diskrepanz zwischen Modell und Wirklichkeit kann zwar durch die Wahl von Markov-Prozessen höherer Ordnung reduziert werden, aber der Preis dafür ist eine exponentiell wachsende Modellkomplexität. So gehen die Vorteile eines Markov-Prozesses erster Ordnung verloren.
Variable Length Markov Chains (VLMC) sind eine Unterklasse von Markov-Prozessen. Diese Modelle können eine zeithomogene Abhängigkeitsstruktur in Daten erkennen und vollständig wiedergeben, ohne dass simultan ihre Komplexität explodiert. Leider basiert das VLMC-Kalkül aber auf Annahmen, welche im Versicherungskontext klar verletzt sind. Zum Beispiel können VLMC zeitliche Abhängigkeiten nicht auf natürliche Art und Weise modellieren.
Mit dem Ziel diese Konflikte zu beheben, verallgemeinern wir das VLMC-Kalkül und entwickeln eine neue Modellklasse: time-inhomogeneous Variable Length Markov Chains (tiVLMC). Wir konstruieren und implementieren einen Algorithmus zur Modellanpassung von tiVLMC an unzensierte als auch an zensierte Daten. Wir diskutieren die Berechnung von Deckungsrückstellungen und stellen konkrete tiVLMC-Modelle für zwei verschiedene Datensätze vor. Der erste Datensatz enthält echte Beobachtungen aus dem Kontext der Pflegeversicherung, der zweite enthält echte Beobachtungen aus dem Kontext einer Versicherung verbundener Leben. Außerdem entwickeln wir Verfahren zur Glättung von tiVLMC-Modellen und stellen so die Erklärbarkeit der Modelle und den reibungslosen Vertrieb des daraus abgeleiteten Versicherungsvertrags sicher.

Contents

List of Figures

List of Tables

List of R-Codes

1 Introduction

1.1 The problem

Many of the manifold tasks of an insurance mathematician, the so-called actuary, consist primarily or at least partly of finding good models to explain relationships between an unknown response and known, observable covariates. To do so, the actuary trains a model of choice on historical data, i.e. past realizations of the response and of the covariates. Then the model allows the prediction of the unknown response for new combinations of covariates, i.e. for new observations. Whole books have been published dealing with this modelling process in general, e.g. Rotar (2014) and De Jong et al. (2008), or covering specific subsectors of insurance, e.g. Haberman and Pitacco (1998) and Lemaire (2013).
As a more concrete example, the pricing of insurance contracts is one major task of an actuary. In order to evaluate the best pricing methods it has attracted a lot of attention within the academic community, cf. e.g. Levikson and Mizrahi (1994), Wang (2002), Embrechts et al. (1996) and many others. In its most basic form an insurance contract can be understood as a legal agreement between two parties: the insured and the insurance company. As part of this legal agreement the insurance company agrees to pay for future expenses S caused by defined events in exchange for a premium π paid by the insured. Usually either the amount S of the future expenses is random and/or it is random whether the defined events occur/how often they occur. On the other hand the paid premium π is non-random, its amount and due date are known by both the insured and the insurance company. By closing an insurance contract the insured therefore trades randomness against certainty, cf. Longley-Cook (1961). This of course comes at a price: The premium π usually is higher than the expected future expenses $\mathbb{E}[S]$, i.e.

$$\pi > \mathbb{E}[S].$$

If we disregard the existence of other costs of the insurance company, like rent, utilities, dividends, interest payments, salaries etc., we can decompose the premium π into a so-called net premium π^{net} and a so-called risk or safety loading π^{risk}, i.e.

$$\pi = \pi^{\text{net}} + \pi^{\text{risk}}.$$

To diversify risks, the net premium π^{net} is in theory determined by the equivalence principle, cf. Norberg (2014): It shall hold that the expected amount $\mathbb{E}[S]$ of the future expenses S equals the net premium π^{net}, i.e.

$$\pi^{\text{net}} = \mathbb{E}[S].$$

At first glance using the net premium π^{net} as the premium π could be considered "fair", because the expenses of the insurance company and the premium the insured pays are in balance:

$$\pi = \pi^{\text{net}} = \mathbb{E}[S].$$

But, and this in fact is a well-known result that can be obtained by applying the law of the iterated logarithm, cf. Theorem 9.5 in Billingsley (1995) or Theorem A.8 in the appendix, the insurance company will face bankruptcy with a probability of 100% as time goes by if it only charges the net premium as the premium, i.e. chooses $\pi^{\text{risk}} = 0$. As a direct consequence the insurance company has to make an effort on how to choose π^{risk}. Commonly a so-called premium principle is used to determine π^{risk}, the most basic one being the "premium with safety loading"-principle, where

$$\pi^{\text{risk}} := \delta\mathbb{E}[S]$$

is chosen with a safety factor $\delta > 0$. Other premium principles, e.g. the "variance"-principle

$$\pi^{\text{risk}} := \delta\text{Var}[S],$$

also make use of higher moments like $\mathbb{E}[S^2]$ of S. For more details on premium principles cf. e.g. Chapter 10.1 in Schmidt (2006) or Heilmann and Schröter (2013). In general the insurance company is mathematically pretty free on how to choose π^{risk}. Supply and demand and/or sales policies often influence the choice of π^{risk} the most. Since the distribution of the future expenses S is unknown, in general one cannot simply calculate $\mathbb{E}[S]$ or higher moments of S and hence the real work is to obtain reliable approximations of them in order to calculate π^{net} and π^{risk}. To do so, the actuary tries to find a fitting mathematical model that explains the relationship between S and observable covariates best. In the context of life insurance commonly used covariates are the age of the insured, her or his health condition at conclusion of the contract, whether she or he is a smoker or a non-smoker etc., in the context of car insurance common covariates are the value of the car, the area the insured lives in etc., cf. e.g. Kwon and Jones (2006),

Verbelen et al. (2018) and Wüthrich (2017). The model of choice first gets fitted to historical data and then inputting the covariates of a new customer is used to approximate the expected future expenses $\mathbb{E}[S]$ caused by the new customer, the variance $\mathrm{Var}[S]$ or other values of interest. Therefore the model enables us to compute an approximation of π^{net}, π^{risk} and thus the premium π.
Calculating the premium of an insurance contract is just one example where the described modelling procedure plays a major role. More examples are the calculation of prospective reserves in life insurance, cf. e.g. Norberg (1991) and Milbrodt and Stracke (1997), the prediction of claim frequencies and of claim severities in non-life insurance, cf. e.g. Garrido et al. (2016) and Frangos and Vrontos (2001), the highly topical prediction of cyber-attacks in cyber-insurance, cf. e.g. Böhme et al. (2010) and Herath and Herath (2011), and many more.
While there are countless applications with different purposes outside of the insurance world, quite a large proportion of the insurance applications share a common overall goal: directly or indirectly predicting the amount of future claims the insurance company will have to cover. In practice, as well as in theory, so-called time-discrete Markov processes with finite state spaces are often the prevalent models of choice, cf. e.g. Hoem (1969) or Lin et al. (2008). The reason for this is quite simple: Markov processes are easy to handle. In practice their usage makes computations easy and in theory they provide a strong basis of well-known mathematical results. A Markov process is a stochastic process that fulfills the so-called Markov property. The name "Markov" hereby derives from the Russian mathematician Andrey Markov, who published his first paper on the topic in 1906, cf. Markov (1906) (English translation). More on the historical aspects of Markov processes can be found in the first chapter of Gagniuc (2017). Informally spoken the Markov assumption says that only the present and not the past is carrying relevant information when predicting the future, i.e.

$$\mathbb{P}\left(\text{future} \,|\, \text{present, past}\right) = \mathbb{P}\left(\text{future} \,|\, \text{present}\right),$$

or, expressed differently, a Markov process has no memory. Translated into mathematical language this intuition forms the following definition:

Definition 1.1. A time-discrete stochastic process $X = (X_t)_{t\in\{1,\dots,T\}}$, $T \in \mathbb{N}$, $T \geq 1$, that lives on a probability space $\left(M^T, \mathscr{P}\left(M^T\right), \mathbb{P}\right)$ with values X_t in a

finite set M is called a time-discrete Markov process of first order if it fulfills the Markov assumption

$$\mathbb{P}\left(X_{t+1} = x_{t+1} \,|\, X_t = x_t, ..., X_1 = x_1\right) = \mathbb{P}\left(X_{t+1} = x_{t+1} \,|\, X_t = x_t\right)$$

for all combinations of $x_1, ..., x_{t+1} \in M$ for which the left-hand side is well-defined, $1 \leq t \leq T-1$.

M^T is the set of all M-valued T-tuples, $\mathscr{P}\left(M^T\right)$ the power set of M^T. We call M the state space of X and its elements $x \in M$ states.
In the real world the Markov assumption is often violated. To demonstrate the reason why, let us look at an example:
As an insurance company we sell third-party vehicle insurance contracts to customers. In order to calculate the premium π, a new customer has to pay for the insurance, as a first step we use a claim frequency model to predict how many accidents this new customer will cause based on the number of accidents he caused in the past. Then we might combine this prediction with the average claim size to obtain π^{net} and then π. In our model X_t mathematically denotes the number of car accidents the insurance holder causes in year t and we want to use the model to predict the number of accidents X_{t+1} caused in the coming year $t+1$ based on the accident history $(X_1, ..., X_t)$. If we decided to use a time-discrete Markov process of first order, as defined in above Definition 1.1, the Markov assumption would state that the number of accidents X_t caused in the current year t includes all the information needed to predict the number of future accidents X_{t+1} in the coming year, i.e. her or his further accident history $(X_1, ..., X_{t-1})$ is totally irrelevant to make this prediction: E.g. it does not matter whether the insurance holder caused eight or zero accidents in year $t-1$. As one intuitively understands, this assumption is not true. It is reasonable to say that one needs to consider an accident history consisting of more than just the most recent year in order to obtain accurate predictions.
For many applications time-discrete Markov processes of first order are therefore too simple to display the dependency between the covariates and the response variable. But, as the declaration “first order” already suggests, we can fix this problem by using time-discrete Markov processes of higher order:

Definition 1.2. A time-discrete stochastic process $X = (X_t)_{t\in\{1,...,T\}}$, $T \in \mathbb{N}$, $T \geq 1$, that lives on a probability space $\left(M^T, \mathscr{P}\left(M^T\right), \mathbb{P}\right)$ with values X_t in a finite set M is called a time-discrete Markov process of k-th order if it fulfills

$$\begin{aligned}&\mathbb{P}\left(X_{t+1} = x_{t+1} \,|\, X_t = x_t, ..., X_{t-k+1} = x_{t-k+1}, ..., X_1 = x_1\right)\\ =&\mathbb{P}\left(X_{t+1} = x_{t+1} \,|\, X_t = x_t, ..., X_{t-k+1} = x_{t-k+1}\right)\end{aligned}$$

for all combinations of $x_1, ..., x_{t+1} \in M$ for which the upper probability is well-defined, $1 \leq k \leq t \leq T-1$.

For $k = 1$ both Definitions 1.1 and 1.2 coincide.
Continuing above example, we could use a higher-order time-discrete Markov process as our model now. E.g. we could assume that considering an accident history of five years is sufficient to make precise predictions and thus use a time-discrete Markov process of order $k = 5$. Now not only the most recent number of accidents X_t, but the accident history $(X_{t-4}, X_{t-3}, X_{t-2}, X_{t-1}, X_t)$ of the five most recent years is considered to predict the number X_{t+1} of future accidents.
However, using higher-order time-discrete Markov processes also comes at a cost: With an increasing order k, higher-order time-discrete Markov processes become exponentially more and more complex. Mathematically more precise: The number of parameters needed to fully characterize the model, i.e. the "degrees of freedom", increases exponentially with the model order k. In the mathematical literature this is known as the "curse of dimensionality", an expression that was coined by and in Bellman (1966). Since the simplicity of a first-order time-discrete Markov process is one of its biggest advantages, using higher-order time-discrete Markov processes and just increasing the model order k is often no acceptable alternative. This problem is also confirmed e.g. in Mächler and Bühlmann (2004, page 436-437).
Therefore it is our goal to find a type of model that somehow lies "in between" a first-order and higher-order time-discrete Markov process. We wish to preserve the advantage of simplicity/interpretability while simultaneously keeping the model complexity limited. To do so, we will transform a higher-order model into a first-order one by coding the information contained in the past that is needed to be known in order to make accurate predictions into the present.
The following well-known Theorem 1.3, cf. e.g. Cox and Miller (1977, page 132), states a simple way on how to transform a higher-order time-discrete Markov process into a first-order time-discrete Markov process. The essence of the theorem is

that the information encoding can be done via an expansion of the state space M.

Theorem 1.3. *A time-discrete Markov process of k-th order $X = (X_t)_{t\in\{1,...,T\}}$ with finite state space M can be transformed to a time-discrete Markov process of first order $X^k = \left(X_t^k\right)_{t\in\{k,...,T\}}$ by setting*

$$X_t^k := (X_t, X_{t-1}, ..., X_{t-k+1})$$

for $t \geq k$. The state space of X^k is M^k.

More precisely, by applying Theorem 1.3, we convert a higher-order time-discrete Markov process into a time-discrete Markov process of first order, by coding all the information from the past into the present. Since the original process was of order k, the k most recent time steps contained all the information. In the transformed setting all the information is contained in the most recent time step, but the state space is now M^k instead of M, if $m := |M|$, its cardinality is $|M^k| = m^k$ now. Therefore this reduction of order does not reduce the complexity of the model, in fact the model complexity remains unchanged: Due to the order decrease from k to one we only need to look at the most recent time point to make our prediction instead of the k most recent time points, but now the most recent time point can take m^k different values while, before the transformation, each of the k most recent time points only could take m different values.
Thus encoding all the information from the past into the present yields no reduction in model complexity and therefore it is not achieving our formulated goal.
We rather need to develop tools that enable us to separate the information contained in the past into two groups: the part of information actually benefiting our predictive power and the part that does not improve our prediction, i.e. the "relevant" and "non-relevant" information contained in the past. If one continues by only coding all the relevant information from the past into the present, then it is intuitive that firstly we achieve the same optimal prediction accuracy as when we encode all the information and secondly the model complexity decreases since we do not encode all the information. Hence we get the best from both perspectives.

1.2 The concept of Variable Length Markov Chains

"Variable Length Markov Chain"-models usually abbreviated by "VLMC" are a model class exactly doing this. VLMC were firstly introduced by and in Ris-

sanen (1983), who mainly worked in the discipline of information theory. They have been subject to extensive further research, major theoretical contributions have been added by Weinberger et al. (1995), Ron et al. (1996), Bühlmann et al. (1999), Ferrari and Wyner (2003), Duarte et al. (2006), Csiszár and Talata (2006), Galves et al. (2008), Xiong et al. (2016) and many more. In their present state of development VLMC-models are great choices for many applications. Research has confirmed their suitability e.g. in click prediction, cf. Borges and Levene (2007) and Gopalakrishnan et al. (2018), protein classification, cf. Bejerano and Yona (2001) or Busch et al. (2009), music classification, cf. Pachet (2002) or Dubnov et al. (2003), linguistic pattern recognition, cf. Galves et al. (2012) or García et al. (2017), just to name a few of many applications from manifold disciplines. But, when it comes to insurance, the assumptions required for VLMC-models to work are violated in the majority of applications. We will point out reasons for this statement in the following Chapter 1.3.

Nevertheless VLMC-models seem to possess big potential when it comes to problem solving in the context of insurance. Especially they are a great tool for data analysis as they expose dependencies in the data in a distinctly clear way. This thesis is devoted to opening up and exploiting this potential. Firstly we are going to end this chapter by giving a compact but sufficient overview of the current VLMC-setup. As a second step we are going to identify the weaknesses of this current VLMC-setup from the viewpoint of insurance in the subsequent Chapter 1.3. In the last introductory Chapter 1.4 we are going to discuss our approach on how to generalize the current VLMC-setup in a way that eliminates the identified weaknesses and thus give a short overview of the thesis.

To present the part of the current VLMC-setup that is relevant for this work, we mainly follow Bühlmann et al. (1999) due to their compact way of writing. While there are many more results available and while one could give many more interpretations and background explanations of the following definitions and statements, we focus on the bare necessities needed to be known to understand the weaknesses of the current VLMC-setup in the following Chapter 1.3. We are going to catch up on the backgrounds and go into deep detail when we generalize the current setup to an insurance setting.

Let $X = (X_t)_{t\in\mathbb{Z}}$ be a stationary, time-discrete stochastic process with values X_t in an unordered state space M of finite cardinality. We emphasize that X here is assumed to be stationary, i.e. it is assumed that

$$\mathbb{P}\left(X_{t_1+t} = x_{t_1}, ..., X_{t_n+t} = x_{t_n}\right) = \mathbb{P}\left(X_{t_1} = x_{t_1}, ..., X_{t_n} = x_{t_n}\right) \tag{1.1}$$

holds for all $t_1, ..., t_n \in \mathbb{Z}$, $n \in \mathbb{N}$ and $t \in \mathbb{N}$.
Within the VLMC-community it has proven useful to use a reversed notation where the most recent time point is notated on the left.

Definition 1.4. For $-\infty \leq k \leq l \leq \infty$ we write

$$X_{k:l} := (X_l, X_{l-1}, ..., X_{k+1}, X_k)$$

and

$$x_{k:l} := (x_l, x_{l-1}, ..., x_{k+1}, x_k) \in M^{l-k+1}.$$

By

$$|x_{k:l}| := l - k + 1$$

we denote the length of the sequence $x_{k:l}$. e is the unique sequence of length zero. For $w \in M^{|w|}$ and $u \in M^{|u|}$ with $0 \leq |w| + |u| \leq T$, we define the concatenation of the two sequences as

$$wu := w \cdot u := (w_{|w|}, ..., w_1, u_{|u|}, ..., u_1) \in M^{|w|+|u|} = M^{|wu|}.$$

Definition 1.5. For $0 \leq k \leq l \leq \infty$ let

$$M^{k:l} := \left\{ w \in M^{|w|} : k \leq |w| \leq l \right\}$$

denote the set of all sequences of length k to l.

The most important object in the VLMC-framework are the so-called "context functions" as they are the tool that differentiates the relevant information contained in the past from the non-relevant information.

Definition 1.6. We call

$$c : M^\infty \to M^\infty,\ x_{-\infty:0} \mapsto x_{-l+1:0}$$

the context function of X, where

$$\begin{aligned} l = l(x_{-\infty:0}) := \min \Big\{ k : \quad & \mathbb{P}\left(X_1 = x_1 \,|\, X_{-\infty:0} = x_{-\infty:0}\right) \\ & = \mathbb{P}\left(X_1 = x_1 \,|\, X_{-k+1:0} = x_{-k+1:0}\right) \text{ for all } x_1 \in M \Big\}. \end{aligned}$$

Recall that the stationarity assumption (1.1) implies that it is sufficient to only look at $X_{-\infty:1}$ in the way that

$$\mathbb{P}\left(X_1 = x_1 \,|\, X_{-k+1:0} = x_{-k+1:0}\right) = \mathbb{P}\left(X_{t+1} = x_1 \,|\, X_{t-k+1:t} = x_{-k+1:0}\right) \tag{1.2}$$

for all $t \in \mathbb{N}$. A time-discrete Markov process that fulfills this property is called "time-homogeneous" and otherwise "time-inhomogeneous".
We cite Ferrari and Wyner (2003, page 461): "[...] The elements of the set

$$\{c\left(x_{-\infty:0}\right) : x_{-\infty:0} \in M^{\infty}\}$$

are called contexts of the process X. The name context derives from the fact, that now the random variable X_1 does no more depend on the full history $x_{-k+1:0}$, as in the case of a [time-discrete] Markov [process] of order k, but only on some pieces of variable length l from the infinite past $x_{-\infty:0}$." In other words, the context is exactly what we called the relevant information from the past.

Definition 1.7. Let $X = (X_t)_{t\in\mathbb{Z}}$ be a stationary, time-discrete stochastic process with values X_t in a finite state space M and corresponding context function c as given in Definition 1.6. If $0 \leq k \leq \infty$ denotes the smallest integer such that

$$\max\left\{\left|c\left(x_{-\infty:0}\right)\right| : x_{-\infty:0} \in M^{\infty}\right\} \leq k$$

is fulfilled we call X a Variable Length Markov Chain (VLMC) of order k.

"A VLMC of order k can be embedded in a [time-discrete] Markov [process] of order k, however with a memory of variable length $l(\cdot) \leq d$. The case $l(\cdot) \equiv 0$ coincides with an independent, stationary process. If $c\left(x_{-\infty:0}\right) = x_{-d+1:0}$ [for all] $x_{-\infty:0} \in M^{\infty}$, then X is a full [time-discrete] Markov [process] of order k. Since there is a large variety of context functions [...] with different structures (particularly of sparse type), VLMC of order k build a more flexible class of processes than full [time-discrete] Markov [processes] of order k, and they better face the curse of dimensionality.", cf. Ferrari and Wyner (2003, page 461-462). Therefore VLMC are a perfect match when it comes to the fulfilment of our formulated goal.
Additionally, they are a great tool to explore dependency structures within data. While we are going to look into the details of this later, the main reason is that context functions can be illustrated as trees. This is due to their hierarchical

nature.

Definition 1.8. Let c be a context function of a VLMC X. The context tree τ of X is defined as

$$\tau := \text{image}(c) = \{c\,(x_{-\infty:0}) : x_{-\infty:0} \in M^{\infty}\}$$

and the terminal node context tree τ^{t} of X as

$$\tau^{\mathsf{t}} := \{w \in \tau : wu \notin \tau \text{ for all } u \in M\}.$$

Definition 1.9. For $x \in M^{1:T}$ let

$$N_{+}(x) := \sum_{t=1}^{T-|x|+1} \mathbb{1}(X_{t-|x|+1:t} = x)$$

denote the number of occurrences of the state change sequence x in $X_{1:T}$ and

$$N(x) := \sum_{t=1}^{T-|x|} \mathbb{1}(X_{t-|x|+1:t} = x)$$

the number of occurrences in $X_{1:T-1}$.

Definition 1.10. For $x \in M$ and $w \in M^{1:\infty}$ with $N(w) > 0$ define the empirical transition probability as

$$\check{P}(x \mid w) := \frac{N_{+}(xw)}{N(w)}.$$

The empirical transition probabilities defined as in Definition 1.10 constitute the maximum likelihood estimate for the true transition probabilities, cf. Chapter 2.2 in Anderson and Goodman (1957).
Rissanen (1983) proposed an algorithm to infer VLMC from data. Since then this algorithm is known in literature under the name “Context”. Context is based on the idea of tree pruning and therefore shares similarities with the CART algorithm developed by and in Breiman et al. (1984). Garivier and Leonardi (2011, page 2489) express the underlying idea as follows: “A measure of discrepancy between a node’s children determines whether they have to be removed from the tree or not.” The measure of discrepancy chosen originally by Rissanen (1983) is

the Kullback-Leibler divergence, cf. Kullback and Leibler (1951), but other measures, e.g. the L^1- or L^∞-norm have been considered as well in Bühlmann et al. (1999) or Galves et al. (2008) respectively.

Algorithm 1.11.
Input: $X_{1:T}^1 = x_{1:T}^1$, *a cutoff* $C > 2|M| + 2$

Step 1: Tree growing
Construct the maximal terminal node context tree $\tau_{\max}^t$ *such that*

$$w \in \tau_{\max}^t \Rightarrow N(w) \geq 2$$

and

$$\forall \tau^t \text{ with } w \in \tau^t \Rightarrow N(w) \geq 2 : \Big(\forall w \in \tau^t \, \exists u \in M^{0:\infty} : wu \in \tau_{\max}^t\Big).$$

Set $\tau_{(0)}^t = \tau_{\max}^t$.

Step 2: Leaf pruning
Examine every element $wu \in \tau_{(0)}^t$, $u \in M$, *as follows:*
Prune wu *down to* w *if*

$$N(wu) \sum_{x \in M} \check{P}(x \mid wu) \log \left(\frac{\check{P}(x \mid wu)}{\check{P}(x \mid w)} \right) < C \log T,$$

else do nothing.
This yields a tree $\tau_{(1)}$.

Step 3: Stopping criterion
Repeat Step 2 with $\tau_{(i)}^t$ *instead of* $\tau_{(i-1)}^t$ *until no more pruning can be done. Call this maximally pruned tree* $\check{\tau}$.

Output: $\check{\tau}$

It has been shown that this estimator $\check{\tau}$ is consistent in the sense of Theorem 1.12, for proof cf. the proof of Theorem 1 in Weinberger et al. (1995, page 648) or Theorem 3.2 in Bühlmann et al. (1999, page 493), which is formulated in a slightly more general setting.

Theorem 1.12. *Let $X = (X_t)_{1 \leq t \leq T}$ be a VLMC with finite state space M and (unknown) context tree τ. If*

$$\min_{x_1 \in M,\, w \in \tau} \mathbb{P}\left(X_1 = x_1 \,\middle|\, X_{-|w|+1:0} = w\right) > 0$$

holds and $\check{\tau}$ is the output of Algorithm 1.11,

$$\lim_{T \to \infty} \mathbb{P}(\check{\tau} = \tau) = 1$$

holds.

1.3 Special requirements for insurance processes

In Chapter 1.2 we pointed out that, at least at first glance, the current VLMC-setup seems to be capable of achieving the goal formulated at the end of Chapter 1.1. Consistent estimators as well as algorithms that allow us to fit VLMC-models to actual data have already been proposed, cf. Algorithm 1.11 and Theorem 1.12. This brings up the question why and how a second, more detailed look at the current VLMC-setup reveals that it is actually not suited to achieve the set goal and therefore constitutes the reason why more work is needed to generalize this setup. The answer to this question mainly lies in the assumptions needed to make the current VLMC-setup work. More precisely, we needed to assume from the very beginning that

(A.1) X is stationary.

To obtain a consistent VLMC-estimator, we also needed to assume that the minimal transition probability of the process is non-negative, i.e.

(A.2) $\min_{x_1 \in M,\, w \in \tau} \mathbb{P}\left(X_1 = x_1 \,\middle|\, X_{-|w|+1:0} = w\right) > 0.$

Both assumptions, (A.1) as well as (A.2), are highly incompatible with many applications from the context of insurance.
The stationarity assumption (A.1) implies that X is time-homogeneous. If an insurance agent models e.g. the development of the health status of an insured, the probabilities of moving from one category to another in general highly depend on the age t of the insured and therefore this constitutes a time-inhomogeneous application. As an even more precise example, the probability of dying, which would correspond to a transition from the health status “alive” to “deceased”

trivially depends on the age t, as on average a healthy 80-year-old is more likely to die within the next period of time compared to an otherwise similar 20-year-old. Hence, even the less strict assumption of time-homogeneity causes a conflict with the viewpoint of insurance. Mathematically, none of the Definitions 1.6-1.10 would make sense anymore. The context function of X would vary at every time t, since the transition probabilities of X would vary with t and equation (1.2) would be invalid. Varying transition probabilities prohibit the summation of the counting frequencies over different time points in Definition 1.9. The empirical transition probabilities, cf. Definition 1.10, would require a reformulation that allows them to vary with the time t as well. Obviously the fitting algorithm Context and his current formulation Algorithm 1.11 are not capable to work in a time-inhomogeneous setting.

The requirement (A.2) of transition probabilities that are truly bound away from zero especially implies that X is ergodic. Informally spoken a stochastic process is called ergodic, if its statistical properties can be fully deduced from a single observation of sufficient length. Ergodicity is a terminus first introduced by and in Boltzmann (1884). (A.2) clearly implies the non-existence of "absorbing states", since the probability to leave the current state always has to be truly positive. Hereby we call a state "absorbing", if the probability to remain in this state is one. This non-existence is also implied by the less strict assumption of ergodicity, since we do not learn anything about the statistical properties of a process after it hit an absorbing state. Many insurance applications require models that factor in absorbing states. E.g., continuing our modelling of health statuses, "deceased" would be an absorbing state, since a deceased insured will never recover and will remain in this state forever. If the insured cancels her or his coverage, "lapse" might be another absorbing state. Hence, even if one only went with the less strict assumption of ergodicity, one would already get into conflict with the viewpoint of insurance. If we do not assume ergodicity, we cannot work with only one observation of the process. Instead we have to recover the power to learn everything about the dependency structure of the process by looking at many observations. In insurance terms this is quite intuitive: We do not only look at one insured to calibrate our model, but rather consider our complete portfolio of eligible customers. Mathematically, the proof of Theorem 1.12 does not only rely heavily on assumption (A.2) (and hence especially on ergodicity), but furthermore the considered asymptotics of $T \to \infty$ do not make any sense.

Lastly, the current VLMC-setup does not allow the process X to be censored. In insurance data is often subject to censoring, i.e. it is unknown in what state X

is at certain times t. A main reason for this is that not all insured enter their contracts at the same age, but also profane causes like documentation errors or data loss lead to censored data.

1.4 An approach

We have pointed out the incapabilities of the current VLMC-setup in an insurance context in the previous Chapter 1.3. The allocation of tasks on how to fix these weaknesses now is straightforward: Within the scope of this thesis we are going to generalize the VLMC-setup, the fitting Algorithm 1.11 and propose important tools required within an insurance context.
In the subsequent Chapters 2 and 3, we are going to generalize all components to be time-dependent and eliminate the need of the stationarity assumption (A.1) and even time-homogeneity. We are going to transform the single-observation setup to a setup that works with n independent observations of X. This will allow us to eliminate the requirement of truly positive transition probabilities or ergodicity, i.e. it will make assumption (A.2) obsolete. This change in perspective will allow us to include absorbing states in our generalized setup.
We are also going to take steps to include censoring into the generalized setup.
Within this generalized setup we are going to construct an algorithm based on Context to fit the generalized models to (now possibly censored) data. We are going to state and prove that this algorithm is consistent in the same way as Context is in the current VLMC-setup, i.e. we are going to prove the equivalent of Theorem 1.12 for our algorithm in our setup. This is done in Chapters 4 and 5 and this represents the major theoretical part of this thesis.
Next we are going to answer the question of how to tune the generalized algorithm in Chapter 6. To be of actual use to a practitioner, the generalized algorithm and the tuning procedure need to be implemented in a programming environment: We are going to develop, document and explain an implementation in the statistical programming language R, cf. R Core Team (2018), in Chapter 7 and then demonstrate the application of the generalized algorithm and the tuning procedure on different generated data sets in Chapter 8.
In Chapter 9 we are going to introduce prospective reserves to our environment and we are going to learn how to use these quantities in order to calculate the net premium π^{net} of an insurance contract.
As we will see, moving to a time-dependent viewpoint increases the effort needed to interpret fitted models. To counteract this, we are going to develop smoothing

techniques that reduce the complexity of the model fits in Chapter 11 and therefore make them easier to communicate.
Last we are going to apply all the developed techniques by fitting tiVLMC-models to real insurance data, in this case from the context of long-term care insurance and joint life insurance, cf. Chapters 10 and 12.

2 Theoretical framework

We start to build up the generalized framework.
Let $X = (X_t)_{t\in\{1,...,T\}}$, $T \in \mathbb{N}$, $T \geq 1$, be a time-discrete stochastic process on the probability space $\left(M^T, \mathscr{P}\left(M^T\right), \mathbb{P}\right)$ with values X_t in an unordered state space M of finite cardinality $m \in \mathbb{N}$ with $m \geq 2$.

Definition 2.1. By P we denote the probability distribution of X on $\mathscr{P}\left(M^T\right)$, i.e.

$$P(B) := \mathbb{P}(X \in B)$$

for $B \in \mathscr{P}\left(M^T\right)$.

From now on let $X^1, ..., X^n$ denote $n \in \mathbb{N}$ independent copies of X.

Definition 2.2. By $\hat{P}$ we denote the empirical probability distribution of X on $\mathscr{P}\left(M^T\right)$, i.e.

$$\hat{P}(B) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left(X^i \in B\right)$$

for $B \in \mathscr{P}\left(M^T\right)$.

Here

$$\mathbb{1}\left(X^i \in B\right) := \begin{cases} 1, & X^i \in B \\ 0, & X^i \notin B. \end{cases}$$

Proposition 2.3. *For every* $B \in \mathscr{P}\left(M^T\right)$

$$\hat{P}(B) \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} P(B).$$

Proof of Proposition 2.3. Since $X^1, ..., X^n$ are independent copies of X, they are independent and identically distributed. In consequence the indicators

$$\mathbb{1}\left(X^1 \in B\right), ..., \mathbb{1}\left(X^n \in B\right)$$

are also independent and identically distributed. The expectation of an indicator is the probability of the indicated event and hence is always in $[0, 1]$ and in particular upper bounded by one:

$$\mathbb{E}\left[\mathbb{1}\left(X \in B\right)\right] = 1 \cdot P(B) + 0 \cdot (1 - P(B)) = 1 \cdot P(B) = \mathbb{P}(X \in B) \leq 1.$$

Hence we can apply the strong law of large numbers A.1 and obtain the claimed convergence

$$\hat{P}(B) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left(X^i \in B\right) \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} \mathbb{E}\left[\mathbb{1}(X \in B)\right] = P(B).$$

□

Proposition 2.3 states the pointwise strong consistency of $\hat{P}$. It is trivial that the empirical probability $\hat{P}(B)$ of an event $B \in \mathscr{P}\left(M^T\right)$ is an unbiased estimator of the true probability $P(B)$.

Proposition 2.4. *For every* $B \in \mathscr{P}\left(M^T\right)$

$$\mathbb{E}\left[\hat{P}(B)\right] = P(B)$$

holds.

Proof of Proposition 2.4.

$$\mathbb{E}\left[\hat{P}(B)\right] = \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left(X^i \in B\right)\right] = \mathbb{E}\left[\mathbb{1}\left(X \in B\right)\right] = P(B)$$

□

Proposition 2.5. *It holds that*

$$\max\left\{\left|\hat{P}(B) - P(B)\right| : B \in \mathscr{P}\left(M^T\right)\right\} \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} 0.$$

Proof of Proposition 2.5. Fix $B \in \mathscr{P}\left(M^T\right)$. Using Proposition 2.3 we know that for all $\varepsilon > 0$ there exists a $n_B \in \mathbb{N}$ such that for all $n \geq n_B$

$$\left|\hat{P}(B) - P(B)\right| < \varepsilon$$

holds $\mathbb{P}$-a.s. Since M is of finite cardinality m, the cardinality of M^T is m^T and the power set $\mathscr{P}\left(M^T\right)$ consists of $2^{m^T} < \infty$ elements, cf. Proposition A.3. Hence, for all $n \geq \max\left\{n_B : B \in \mathscr{P}\left(M^T\right)\right\}$, we have that for all $B \in \mathscr{P}\left(M^T\right)$

$$\left|\hat{P}(B) - P(B)\right| < \varepsilon$$

$\mathbb{P}$-a.s. and in particular it holds $\mathbb{P}$-a.s. that

$$\max\left\{\left|\hat{P}(B) - P(B)\right| : B \in \mathscr{P}\left(M^T\right)\right\} < \varepsilon.$$

□

Proposition 2.5 states the uniform strong consistency of $\hat{P}$ and hence is a stronger version of Proposition 2.3.

Definition 2.6. For $1 \leq t \leq T-1$, $x \in M$, $w \in M^{1:t}$ with $\mathbb{P}\left(X_{t-|w|+1:t} = w\right) > 0$ we write

$$P_t(x \mid w) := \mathbb{P}\left(X_{t+1} = x \,\middle|\, X_{t-|w|+1:t} = w\right).$$

$P_t(x \mid w)$ is the probability that the next state change is into x coming from w at times $t - |w| + 1$ to t.

Definition 2.7. For $1 \leq t \leq T$, $w \in M^{1:t}$ with $\mathbb{P}\left(X_{t-|w|+1:t} = w\right) > 0$ let

$$C_{t,\,w} := \left\{x_{1:T} \in M^T : x_{t-|w|+1:t} = w\right\} \in \mathscr{P}\left(M^T\right)$$

denote the set of all sequences in M^T that move through w from $t - |w| + 1$ to t.

It clearly holds that

$$P_t(x \mid w) = \frac{\mathbb{P}\left(X_{t+1:t-|w|+1} = xw\right)}{\mathbb{P}\left(X_{t-|w|+1:t} = w\right)} = \frac{P\left(C_{t+1,\,xw}\right)}{P\left(C_{t,\,w}\right)}.$$

In an analogue way we define the empirical transition probabilities of X.

Definition 2.8. For $1 \leq t \leq T-1$, $x \in M$, $w \in M^{1:t}$ with $\hat{P}\left(C_{t,\,w}\right) > 0$ we call

$$\hat{P}_t(x \mid w) := \frac{\hat{P}\left(C_{t+1,\,xw}\right)}{\hat{P}\left(C_{t,\,w}\right)}$$

the empirical transition probability of moving to x from w at t.

It is well-known (e.g. cf. Anderson and Goodman (1957, page 92) and take a look at the following Proposition 2.11 for a better comparability) that the empirical transition probabilities defined as in Definition 2.8 constitute the maximum likelihood estimates of the true transition probabilities defined in Definition 2.6. This is the reason why we use exactly this definition. It shall also be mentioned that there are alternative estimators available for the transition probabilities with different advantages and disadvantages, e.g. cf. Galves et al. (2008, page 261). For simplicity, we only work with one, namely above estimator here.
Now we prove that the empirical transition probabilities of X preserve the uniform strong consistency stated in Proposition 2.5.

Theorem 2.9. *It holds that*

$$\max \left| \hat{P}_t(x \mid w) - P_t(x \mid w) \right| \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} 0,$$

where the maximum runs over all well-defined transition probabilities, i.e. over $1 \leq t \leq T-1$, $x \in M$, $w \in M^{1:t}$ *with* $\mathbb{P}\left(X_{t-|w|+1:t} = w\right) > 0$.

Proof of Theorem 2.9. We formalize

$$\begin{aligned} a &:= \hat{P}(C_{t+1,\,xw}) \\ b &:= \hat{P}(C_{t,\,w}) \\ c &:= P(C_{t+1,\,xw}) \\ d &:= P(C_{t,\,w}). \end{aligned}$$

Then

$$\begin{aligned} \max \left| \frac{a}{b} - \frac{c}{d} \right| = \max \left| \frac{ad-bc}{bd} \right| &= \max \left| \frac{(ad-ab)+(ab-bc)}{bd} \right| \\ &\leq \max \left| \frac{a(d-b)}{bd} \right| + \max \left| \frac{b(a-c)}{bd} \right| \\ &\leq \frac{\max |a(d-b)|}{\min |bd|} + \frac{\max |b(a-c)|}{\min |bd|}, \end{aligned}$$

where the ranges of the minimums and the maximums are the range specified in the theorem.

a and b are upper bounded by one. Proposition 2.3 and $\mathbb{P}\left(X_{t-|w|+1:t} = w\right) > 0$ imply

$$b \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} d > 0,$$

Proposition 2.5 and $\mathbb{P}\left(X_{t-|w|+1:t} = w\right) > 0$ imply that

$$1 \geq \min |bd| > \varepsilon > 0$$

$\mathbb{P}$-a.s. for n large enough. Using that

$$\max |d - b| \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} 0$$

and

$$\max |a - c| \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} 0$$

by Proposition 2.5, implies

$$\max \left|\frac{a}{b} - \frac{c}{d}\right| \leq \frac{1}{\varepsilon} \max |1 \cdot (d - b)| + \frac{1}{\varepsilon} \max |1 \cdot (a - c)| \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} 0.$$

□

Definition 2.10. For $w \in M^{1:t}$ and $|w| \leq t \leq T$ let

$$N_t^i(w) := \mathbb{1}\left(X_{t-|w|+1:t}^i = w\right)$$

indicate whether X^i has moved through w till t, $i = 1, ..., n$. Summarizing over $i = 1, ..., n$,

$$N_t(w) := \sum_{i=1}^{n} N_t^i(w)$$

denotes the number of observations that have moved through w till t.

Using these counting quantities we can rewrite the empirical transition probabilities:

Proposition 2.11. *For $1 \leq t \leq T-1$, $x \in M$, $w \in M^{1:t}$ with $N_t(w) > 0$ it holds that*

$$\hat{P}_t(x \mid w) = \frac{N_{t+1}(xw)}{N_t(w)}.$$

Proof of Proposition 2.11. For arbitrary $z \in M^{1:t}$

$$\hat{P}(C_{t,z}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left(X^i \in C_{t,z}\right) = \frac{1}{n}\sum_{i=1}^{n} N_t^i(z) = \frac{1}{n}N_t(z).$$

Thus

$$\hat{P}_t(x \mid w) = \frac{\hat{P}(C_{t+1,xw})}{\hat{P}(C_{t,w})} = \frac{\frac{1}{n}N_{t+1}(xw)}{\frac{1}{n}N_t(w)} = \frac{N_{t+1}(xw)}{N_t(w)}.$$

□

While the unbiasedness of the empirical probability is trivial, cf. Proposition 2.4, it is not clear whether the empirical transition probabilities are bias-free estimators. The reason for this is that the expectation of a product of dependent random variables does in general not split up.

Theorem 2.12. *For $1 \leq t \leq T-1$, $x \in M$, $w \in M^{1:t}$ with $N_t(w) > 0$ it holds that*

$$\mathbb{E}\left[\hat{P}_t(x \mid w)\right] = P_t(x \mid w).$$

Proof of Theorem 2.12. We start by expressing the empirical transition probabilities as relative frequencies:

$$\mathbb{E}\left[\hat{P}_t(x \mid w)\right] = \mathbb{E}\left[\frac{N_{t+1}(xw)}{N_t(w)}\right] = \mathbb{E}\left[\frac{\sum_{i=1}^{n} N_{t+1}^i(xw)}{N_t(w)}\right].$$

Now note that

$$\begin{aligned} N_{t+1}^i(xw) = \mathbb{1}\left(X_{t-|xw|+2:t+1}^i = xw\right) &= \mathbb{1}\left(X_{t+1}^i = x\right) \cdot \mathbb{1}\left(X_{t-|w|+1:t}^i = w\right) \\ &= \mathbb{1}\left(X_{t+1}^i = x\right) \cdot \mathbb{1}\left(X_{t-|w|+1:t}^i = w\right)^2 \\ &= N_{t+1}^i(xw) \cdot N_t^i(w). \end{aligned}$$

Because $X^1, ..., X^n$ are identically distributed, the same is true for

$$N_{t+1}^1(xw) \cdot N_t^1(w), ..., N_{t+1}^n(xw) \cdot N_t^n(w)$$

and thus

$$\mathbb{E}\left[\hat{P}_t(x \mid w)\right] = n\mathbb{E}\left[\frac{N_{t+1}^1(xw) \cdot N_t^1(w)}{N_t(w)}\right].$$

Now we factorize the expectation:

$$\begin{aligned}
\mathbb{E}\left[\hat{P}_t(x \mid w)\right] =& \sum_{j=1}^{n} n\mathbb{E}\left[\frac{1 \cdot N_{t+1}^1(xw)}{j} \,\middle|\, N_t(w) = j,\ N_t^1(w) = 1\right] \\
& \cdot \mathbb{P}\left(N_t^1(w) = 1 \,\middle|\, N_t(w) = j\right) \mathbb{P}(N_t(w) = j) \\
=& \sum_{j=1}^{n} \frac{n}{j} \mathbb{E}\left[N_{t+1}^1(xw) \,\middle|\, N_t^1(w) = 1\right] \\
& \cdot \mathbb{P}\left(N_t^1(w) = 1 \,\middle|\, N_t(w) = j\right) \mathbb{P}(N_t(w) = j) \\
=& \sum_{j=1}^{n} \frac{n}{j} \mathbb{P}\left(X_{t+1} = x \,\middle|\, X_{t-|w|+1:t} = w\right) \\
& \cdot \mathbb{P}\left(N_t^1(w) = 1 \,\middle|\, N_t(w) = j\right) \mathbb{P}(N_t(w) = j) \\
=& \sum_{j=1}^{n} \frac{n}{j} P_t(x \mid w) \mathbb{P}\left(N_t^1(w) = 1 \,\middle|\, N_t(w) = j\right) \mathbb{P}(N_t(w) = j) \\
=& P_t(x \mid w) \sum_{j=1}^{n} \frac{n}{j} \mathbb{P}\left(N_t^1(w) = 1 \,\middle|\, N_t(w) = j\right) \mathbb{P}(N_t(w) = j)
\end{aligned}$$

Now

$$N_t^1(w) \sim \text{Bin}\left(1, \mathbb{P}\left(X_{t-|w|+1:t} = w\right)\right)$$

and

$$N_t(w) - N_t^1(w) \sim \text{Bin}\left(n-1, \mathbb{P}\left(X_{t-|w|+1:t} = w\right)\right).$$

Thus

$$\begin{aligned}
& \mathbb{P}\left(N_t^1(w) = 1 \,\middle|\, N_t(w) = j\right) \\
=& \frac{\mathbb{P}\left(N_t^1(w) = 1,\ N_t(w) = j\right)}{\mathbb{P}\left(N_t(w) = j\right)} \\
=& \frac{\mathbb{P}\left(N_t^1(w) = 1\right) \mathbb{P}\left(N_t(w) - N_t^1(w) = j - 1\right)}{\mathbb{P}\left(N_t(w) = j\right)} \\
=& \frac{\mathbb{P}\left(N_t^1(w) = 1\right) \binom{n-1}{j-1} \mathbb{P}\left(N_t^1(w) = 1\right)^{j-1} \left(1 - \mathbb{P}\left(N_t^1(w) = 1\right)\right)^{n-j}}{\binom{n}{j} \mathbb{P}\left(N_t^1(w) = 1\right)^{j} \left(1 - \mathbb{P}\left(N_t^1(w) = 1\right)\right)^{n-j}} \\
=& \frac{(n-1)!}{n!} \frac{j!}{(j-1)!} \\
=& \frac{j}{n}.
\end{aligned}$$

To obtain the second equality, we used that $N_t^1(w)$ and

$$N_t(w) - N_t^1(w) = \sum_{i=2}^{n} N_t^i(w)$$

are independent, since $X^1, ..., X^n$ are. The unbiasedness of the empirical transition probabilities follows:

$$\mathbb{E}\left[\hat{P}_t(x \mid w)\right] = P_t(x \mid w) \sum_{j=1}^{n} \frac{n}{j} \frac{j}{n} \mathbb{P}(N_t(w) = j) = P_t(x \mid w) \cdot 1 = P_t(x \mid w).$$

□

Being strongly consistent and unbiased, the empirical transition probabilities defined in Definition 2.8 are reasonable estimators of the true transition probabilities.

3 Time-inhomogeneous Variable Length Markov Chains

Now we continue by generalizing the current setup of VLMC as presented in Chapter 1.2. The major extension is the switch from the time-homogeneous environment, recall the special requirements for insurance processes discussed in Chapter 1.3 and more precisely (A.1), to a time-inhomogeneous environment. Therefore we call the generalized models "time-inhomogeneous Variable Length Markov Chains", abbreviated by "tiVLMC".
Since we do not assume that all transition probabilities are truly positive, i.e. we do not assume (A.2), there are state change sequences that cannot occur in our generalization. Therefore we will have to be cautious when it comes to the choice of the domain of a generalized, time-dependent context function.

Definition 3.1. For $1 \leq t \leq T-1$ let

$$\operatorname{tpsupp}_t(X) := \left\{ x \in M^t : \mathbb{P}\left(X_{1:t} = x\right) > 0 \right\}$$

denote the transition probability support of X at t.

Example 3.2. Let $M = \{a, i, i_+, d\}$, where a stands for active, i for invalid, i_+ for heavily invalid and d for deceased. Assume that the initial distribution of X is $\mathbb{P}^{X_1} = \delta_a$, i.e. X starts in a with probability one. Then

$$\operatorname{tpsupp}_1(X) = \{a\}.$$

Next assume that the following rules apply for the first transition: Moving from a to i and moving from a to i_+ ("becoming (heavily) invalid"), moving from a to d ("dying") and moving from a to a ("staying active") has a positive probability. Moving from d to a, i or i_+ ("resurrection") has probability zero, therefore staying in d ("staying dead") has probability one. Moving from i or i_+ back to a ("reactivation") has a positive probability. Then

$$\operatorname{tpsupp}_2(X) = \{(a,a), (i,a), (i_+,a), (d,a)\}.$$

Remember that we use a reversed notation.

We can now formulate a time-dependent generalization of Definition 1.6 and then continue by formally introducing tiVLMC.

Definition 3.3. For $1 \leq t \leq T-1$ we call

$$c_t : \mathrm{tpsupp}_t(X) \to M^{0:t},\ x_{1:t} \mapsto x_{t-l_t+1:t}$$

the context function of X at t, where

$$l_t = l_t(x_{1:t}) := \min\Big\{0 \leq k \leq t : P_t(x \mid x_{1:t}) = P_t(x \mid x_{t-k+1:t}) \text{ for all } x \in M\Big\}.$$

We further call

$$|c_t| := \max_{x \in \mathrm{tpsupp}_t(X)} l_t(x)$$

the order of the context function at t and

$$c := (c_t)_{1 \leq t \leq T-1}$$

is simply called the context function of X.

Note that $l := (l_t)_{1 \leq t \leq T-1}$ determines c and vice versa. The case $l_t \equiv 0$ corresponds to independence, i.e. to $P_t(x \mid w) = \mathbb{P}(X_{t+1} = x)$ for all $w \in \mathrm{tpsupp}_t(X)$.

Example 3.4. We choose $T \geq 4$ here. Obeying the transition rules of Example 3.2, we have

$$\mathbb{P}\left(X_1 = a\right) = 1,$$

for the first transition let

$$\begin{aligned} P_1(a \mid a) &= 0.7, \\ P_1(i \mid a) &= 0.2, \\ P_1(i_+ \mid a) &= 0.05, \\ P_1(d \mid a) &= 0.05. \end{aligned}$$

Then the context function at $t = 1$ is

$$c_1 : \{a\} \to M^{0:1},\ a \mapsto e.$$

Because X_1 equals a with probability one, this information is useless when it comes to predicting X_2. For the second transition let the transition probabilities be

$$
\begin{aligned}
&P_2(a \mid (a,a)) = P_2(a \mid a) = 0.7,\\
&P_2(i \mid (a,a)) = P_2(i \mid a) = 0.2,\\
&P_2(i_+ \mid (a,a)) = P_2(i_+ \mid a) = 0.05,\\
&P_2(d \mid (a,a)) = P_2(d \mid a) = 0.05
\end{aligned}
$$

$$
\begin{aligned}
&P_2(a \mid (i,a)) = P_2(a \mid i) = 0.05,\\
&P_2(i \mid (i,a)) = P_2(i \mid i) = 0.7,\\
&P_2(i_+ \mid (i,a)) = P_2(i_+ \mid i) = 0.2,\\
&P_2(d \mid (i,a)) = P_2(d \mid i) = 0.05,
\end{aligned}
$$

$$
\begin{aligned}
&P_2(a \mid (i_+,a)) = P_2(a \mid i_+) = 0.05,\\
&P_2(i \mid (i_+,a)) = P_2(i \mid i_+) = 0.2,\\
&P_2(i_+ \mid (i_+,a)) = P_2(i_+ \mid i_+) = 0.05,\\
&P_2(d \mid (i_+,a)) = P_2(d \mid i_+) = 0.7,
\end{aligned}
$$

$$
\begin{aligned}
&P_2(a \mid (d,a)) = P_2(a \mid d) = 0,\\
&P_2(i \mid (d,a)) = P_2(i \mid d) = 0,\\
&P_2(i_+ \mid (d,a)) = P_2(i_+ \mid d) = 0,\\
&P_2(d \mid (d,a)) = P_2(d \mid d) = 1.
\end{aligned}
$$

Then the context function at $t = 2$ is

$$
c_2 : \{(a,a),(i,a),(i_+,a),(d,a)\} \to M^{0:2},\ x_{1:2} \mapsto \begin{cases} a, & x_2 = a \\ i, & x_2 = i \\ i_+, & x_2 = i_+ \\ d, & x_2 = d. \end{cases}
$$

For $t \geq 3$, we set the transition probabilities as following:

$$
\begin{aligned}
P_t(a \mid a \cdot x_{1:t-1}) &= P_t(a \mid a) = 0.7,\\
P_t(i \mid a \cdot x_{1:t-1}) &= P_t(a \mid a) = 0.2,\\
P_t(i_+ \mid a \cdot x_{1:t-1}) &= P_t(a \mid a) = 0.05,\\
P_t(d \mid a \cdot x_{1:t-1}) &= P_t(a \mid a) = 0.05,
\end{aligned}
$$

for $a \cdot x_{1:t-1} \in \mathrm{tpsupp}_t(X)$,

$$
\begin{aligned}
P_t(a \mid i \cdot x_{1:t-1}) &= P_t(a \mid i) = 0.05,\\
P_t(i \mid i \cdot x_{1:t-1}) &= P_t(i \mid i) = 0.7,\\
P_t(i_+ \mid i \cdot x_{1:t-1}) &= P_t(i_+ \mid i) = 0.2,\\
P_t(d \mid i \cdot x_{1:t-1}) &= P_t(d \mid i) = 0.05,
\end{aligned}
$$

for $i \cdot x_{1:t-1} \in \mathrm{tpsupp}_t(X)$,

$$
\begin{aligned}
P_t(a \mid d \cdot x_{1:t-1}) &= P_t(a \mid d) = 0,\\
P_t(i \mid d \cdot x_{1:t-1}) &= P_t(i \mid d) = 0,\\
P_t(i_+ \mid d \cdot x_{1:t-1}) &= P_t(i_+ \mid d) = 0,\\
P_t(d \mid d \cdot x_{1:t-1}) &= P_t(d \mid d) = 1,
\end{aligned}
$$

for $d \cdot x_{1:t-1} \in \mathrm{tpsupp}_t(X)$ and

$$
\begin{aligned}
P_t(a \mid (i_+, i_+) \cdot x_{1:t-2}) &= P_t(a \mid (i_+, i_+)) = 0.25,\\
P_t(i \mid (i_+, i_+) \cdot x_{1:t-2}) &= P_t(i \mid (i_+, i_+)) = 0.05,\\
P_t(i_+ \mid (i_+, i_+) \cdot x_{1:t-2}) &= P_t(i_+ \mid (i_+, i_+)) = 0.4,\\
P_t(d \mid (i_+, i_+) \cdot x_{1:t-2}) &= P_t(d \mid (i_+, i_+)) = 0.4,
\end{aligned}
$$

for $(i_+, i_+) \cdot x_{1:t-2} \in \mathrm{tpsupp}_t(X)$ and, last

$$
\begin{aligned}
P_t(a \mid (i_+, w) \cdot x_{1:t-2}) &= P_t(a \mid i_+) = 0.05,\\
P_t(i \mid (i_+, w) \cdot x_{1:t-2}) &= P_t(i \mid i_+) = 0.2,\\
P_t(i_+ \mid (i_+, w) \cdot x_{1:t-2}) &= P_t(i_+ \mid i_+) = 0.05,\\
P_t(d \mid (i_+, w) \cdot x_{1:t-2}) &= P_t(d \mid i_+) = 0.7,
\end{aligned}
$$

for $(i_+, w) \cdot x_{1:t-2} \in \text{tpsupp}_t(X)$ with $w \neq i_+$. Then the context function at $t \geq 3$ is

$$c_t : \text{tpsupp}_t(X) \to M^{0:t},\ c_t(x_{1:t}) = \begin{cases} a, & x_t = a, \\ i, & x_t = i, \\ i_+, & x_t = i_+,\ x_{t-1} \neq i_+, \\ (i_+, i_+), & x_t = i_+, x_{t-1} = i_+, \\ d, & x_t = d. \end{cases}$$

I.e. the information whether $X_t = a$, $X_t = i$, $X_{t-1:t} = (i_+, w)$ with $w \neq i_+$, $X_{t-1:t} = (i_+, i_+)$ or $X_t = d$ is sufficient to predict X_{t+1}. Further knowledge of the past gives no information benefit.

Definition 3.5. Let $0 \leq k \leq T-1$ be the smallest integer such that

$$\max_{1 \leq t \leq T-1} |c_t| \leq k.$$

Then we call X a time-inhomogeneous Variable Length Markov Chain (tiVLMC) of order k.

Example 3.6. The process X described in Example 3.4 is a tiVLMC of order two.

As it is true for VLMC and time-discrete Markov processes, cf. Chapter 2.1 in Bühlmann et al. (1999) and Frigessi and Heidergott (2011, page 773) respectively, tiVLMC can also be fully described via their transition probabilities and marginal distribution. We state and prove this after the following two definitions.

Definition 3.7. Define the set of all tiVLMC of order k to l, $0 \leq k \leq l \leq T-1$, as

$$\mathcal{P}^{k:l} := \bigcup_{j=k}^{l} \mathcal{P}^j := \bigcup_{j=k}^{l} \{X : X \text{ is a tiVLMC of order } j\}.$$

Definition 3.8. The set of transition probabilities of a tiVLMC $X \in \mathcal{P}^{0:T-1}$ with context function c is

$$\mathbb{P}_c := \{P_t(x \mid c_t(w)) : x \in M,\ 1 \leq t \leq T-1,\ w \in \text{tpsupp}_t(X)\}.$$

Proposition 3.9. *Every tiVLMC $X \in \mathcal{P}^{0:T-1}$ is uniquely determined by its marginal distribution $\mathbb{P}^{X_1}$ and its set of transition probabilities $\mathbb{P}_c$.*

Proof of Proposition 3.9. For every $x_{1:T} \in \mathrm{tpsupp}_T(X)$ it holds that

$$\begin{aligned}
\mathbb{P}\left(X_{1:T} = x_{1:T}\right) &= \mathbb{P}\left(X_T \,|\, X_{1:T-1} = x_{1:T-1}\right) \cdot \mathbb{P}\left(X_{1:T-1} = x_{1:T-1}\right) \\
&= \dots \\
&= \mathbb{P}(X_1 = x_1) \cdot \prod_{t=1}^{T-1} \mathbb{P}\left(X_{t+1} = x_{t+1} \,|\, X_{1:t} = x_{1:t}\right) \\
&= \mathbb{P}(X_1 = x_1) \cdot \prod_{t=1}^{T-1} P_t\left(x_{t+1} \,|\, x_{1:t}\right) \\
&= \mathbb{P}(X_1 = x_1) \cdot \prod_{t=1}^{T-1} P_t\left(x_{t+1} \,|\, c_t(x_{1:t})\right). \qquad (3.1)
\end{aligned}$$

□

The sequences $c_t(x_{1:t})$ determining the transition probabilities of a tiVLMC at t are the values of the context function c_t at t and hence the values of c_t can be interpreted as the minimal state space of the tiVLMC at t. Due to the hierarchical structure of a context function at time t, we can illustrate its image as a tree.

Definition 3.10. For a tiVLMC $X \in \mathcal{P}^{0:T-1}$ with context function c and for $1 \leq t \leq T-1$ we define the context tree τ_t at t by

$$\tau_t := \mathrm{image}(c_t) = \{c_t(w) : w \in \mathrm{tpsupp}_t(X)\}$$

and the terminal node context tree τ_t^{t} at t by

$$\tau_t^{\mathsf{t}} := \{w \in \tau_t : wu \notin \tau_t \text{ for all } u \in M\}.$$

We call $\tau := (\tau_t)_{1 \leq t \leq T-1}$ the context tree of X and $\tau^{\mathsf{t}} := (\tau_t^{\mathsf{t}})_{1 \leq t \leq T-1}$ the terminal node context tree of X.

A (terminal) context tree at t can be illustrated as a hierarchic tree, where the empty string e is the root node and the terminal contexts, i.e. the elements of the terminal context tree, are the branches.

Example 3.11. Continuing with Example 3.6, we get that

$$\tau_1 = \{e\}, \tau_2 = \{a, i, i_+, d\}, \tau_t = \{a, i, i_+, (i_+, i_+), d\}$$

for $t \geq 3$. The terminal context tree for $t \geq 3$ is

$$\tau_t^{\mathsf{t}} = \{a, i, (i_+, i_+), d\}.$$

We can illustrate τ as shown in the following Figure 3.1.

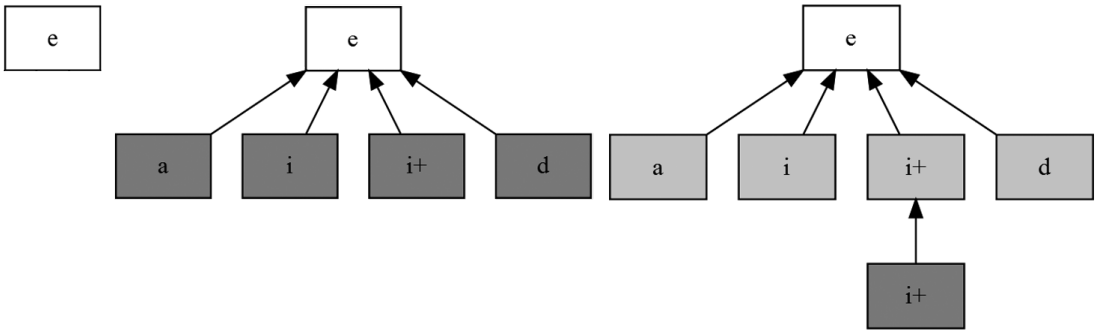

Figure 3.1: Illustration of the context trees τ_1, τ_2 and τ_t for $t \geq 3$ (from left to right)

Note that a context function c_t at t can be reconstructed from the context tree τ_t at t and vice versa. Thus we can also interpret the context tree τ_t at t of a tiVLMC as its minimal state space at t. τ_t can be reconstructed from the terminal node context tree τ_t^{t} at t by just adding the internal nodes to τ_t^{t}. Hence τ_t^{t} and c_t can also be reconstructed from each other.
Shortening a terminal context leads to unequal transition probabilities. This behaviour is asymptotically preserved by the empirical transition probabilities:

Lemma 3.12. *Let $X \in \mathcal{P}^{0:T-1}$ be a tiVLMC with context tree τ and $2 \leq t \leq T-1$. For every $wu \in \tau_t^{\mathsf{t}}$ with $u \in M$ there exist a state $x_{wu} \in M$ with*

$$\varepsilon_{wu} := P_t(x_{wu} \mid wu) - P_t(x_{wu} \mid w) > 0$$

and a threshold $n_{wu} \in \mathbb{N}$ such that $x_{wu} \in A$ for all $n \geq n_{wu}$, where

$$A := \Big\{x \in M : \hat{P}_t(x \mid wu) > \hat{P}_t(x \mid w)\Big\}.$$

Proof of Lemma 3.12. Let $wu \in \tau_t^{\mathsf{t}}$. Then there must exist a state $x_{wu} \in M$ with $\varepsilon_{wu} > 0$. Otherwise

$$P_t(x \mid wu) = P_t(x \mid w)$$

holds for all $x \in M$ and this contradicts $wu \in \tau_t^{\mathsf{t}}$. We can write, by adding zero two times,

$$\begin{aligned}\varepsilon_{wu} =& P_t(x_{wu} \mid wu) - P_t(x_{wu} \mid w) \\ =& \Big(P_t(x_{wu} \mid wu) - \hat{P}_t(x_{wu} \mid wu)\Big) + \Big(\hat{P}_t(x_{wu} \mid wu) - \hat{P}_t(x_{wu} \mid w)\Big) \\ &+ \Big(\hat{P}_t(x_{wu} \mid w) - P_t(x_{wu} \mid w)\Big).\end{aligned}$$

Now, using Theorem 2.9, we get

$$\varepsilon_{wu} = 0 + \lim_{n\to\infty} \Big(\hat{P}_t(x_{wu} \mid wu) - \hat{P}_t(x_{wu} \mid w)\Big) + 0.$$

Thus there exists a threshold integer n_{wu} such that for all $n \geq n_{wu}$ we have

$$\hat{P}_t(x_{wu} \mid wu) > \hat{P}_t(x_{wu} \mid w)$$

and thus $x_{wu} \in A$. This completes the proof. □

If the true transition probabilities in Lemma 3.12 are unequal and at the same time the empirical transition probabilities are close to each other, it must be that at least one empirical probability is way off its true value:

Lemma 3.13. *Let $X \in \mathcal{P}^{0:T-1}$ be a tiVLMC with context tree τ and $2 \leq t \leq T-1$. Let $wu \in \tau_t^{\mathsf{t}}$ and $x_{wu} \in M$ with $\varepsilon_{wu} > 0$. If*

$$\hat{P}_t(x_{wu} \mid wu) - \hat{P}_t(x_{wu} \mid w) < \gamma$$

holds for all $\gamma < \frac{\varepsilon_{wu}}{2}$, it must be that either

$$\Big|\hat{P}_t(x_{wu} \mid wu) - P_t(x_{wu} \mid wu)\Big| > \sqrt{\xi}$$

or

$$\Big|\hat{P}_t(x_{wu} \mid w) - P_t(x_{wu} \mid w)\Big| > \sqrt{\xi},$$

where

$$\xi := \left(\frac{\varepsilon_{wu}}{2} - \gamma\right)^2.$$

We add to the sketch proof of this Lemma by Bühlmann et al. (1999, page 502-503):

Proof of Lemma 3.13. Let $wu \in \tau_t^{\mathfrak{t}}$ and $x_{wu} \in M$ with $\varepsilon_{wu} > 0$. We formalize the claim by defining

$$\begin{aligned} a &:= \hat{P}_t(x_{wu} \mid wu) \\ b &:= \hat{P}_t(x_{wu} \mid w) \\ r &:= P_t(x_{wu} \mid wu) \\ s &:= P_t(x_{wu} \mid w). \end{aligned}$$

Then the claim translates to the following:
If $|a-b| < \gamma$ it must be that either $|a-r| > \sqrt{\xi}$ or $|b-s| > \sqrt{\xi}$.
We have that $r - s = \varepsilon_{wu} > 0$. Assume that $|a-b| < \gamma$.
1. case: $b < s$
Applying the reversed triangle inequality and using $r > s > b$ yields

$$\begin{aligned} |a-r| = |r-a| = |r-b+b-a| &\geq |r-b| - |b-a| \\ &= r - b - |a-b| \\ &> r - s - |a-b| \\ &> \varepsilon_{wu} - \gamma \\ &> \sqrt{\xi}. \end{aligned}$$

2. case: $b < r$
It holds that

$$|b-s| = b - s > r - s = \varepsilon_{wu} > \sqrt{\xi}.$$

3. case: $s \leq b \leq r$
3.1 case: $s \leq b \leq s + \frac{r-s}{2}$, i.e. b is closer (or equally far away) to s than to r
Again applying the reversed triangle inequality yields

$$\begin{aligned} |a-r| = |r-b+b-a| &\geq |r-b| - |b-a| \\ &= r - b - |a-b| \\ &\geq r - \left(s + \frac{r-s}{2}\right) - |a-b| \\ &= \frac{r-s}{2} - |a-b| \end{aligned}$$

$$> \frac{\varepsilon_{wu}}{2} - \gamma$$
$$= \sqrt{\xi}.$$

3.2 case: $r - \frac{r-s}{2} \leq b \leq r$, i.e. b is closer (or equally far away) to r than to s
It holds that

$$|b-s| = b - s > r - \frac{r-s}{2} - s = \frac{r-s}{2} = \frac{\varepsilon_{wu}}{2} > \sqrt{\xi}.$$

This completes the proof. □

Proposition 3.14. *Every tiVLMC $X \in \mathcal{P}^k$ of order $k \in \mathbb{N}_0$ is a time-discrete Markov process of order k.*

Proof of Proposition 3.14. We can embed a tiVLMC of order k into a k-th-order time-discrete Markov process: For $t \geq k$ it holds that

$$P_t\left(x_{t+1} \,|\, x_{1:t}\right) = P_t\left(x_{t+1} \,|\, c_t(x_{1:t})\right) = P_t(x_{t+1} \mid x_{t-k+1:t}),$$

because $k = \max\{|c_t(w)| : 1 \leq t \leq T-1,\ w \in \mathrm{tpsupp}_t(X)\}$. By construction there exist at least one t and one $w \in \mathrm{tpsupp}_t(X)$ with $|c_t(w)| = k$, thus the order of the time-discrete Markov process is not smaller than k. Hence the order is k. The transition probabilities are given by

$$P_t\left(x_{t+1} \,|\, x_{t-k+1:t}\right) = P_t\left(x_{t+1} \,|\, c_t(x_{1:t})\right)$$

and the initial distribution is $\mathbb{P}^{X_{1:k}}$. □

The embedding time-discrete Markov process and the tiVLMC of order k coincide if, for every $t \geq k$, the context function at t is the "full" projection, i.e. it maps $x_{1:t} \mapsto x_{t-k+1:t}$. If this is not the case, the tiVLMC is able to describe the same dependency structure as the time-discrete embedding Markov process while having less model parameters. This is one of two big advantages tiVLMC possess over their embedding. A time-inhomogeneous, time-discrete Markov process $X = (X_t)_{1 \leq t \leq T}$ of order k on M with $|M| = m$ has

$$\gamma(k, m, T) = (m^k - 1) + (m-1)m^k(T-k) \tag{3.2}$$

model parameters. This formula can be derived as follows: $(m^k - 1)$ parameters are needed to describe the initial distribution $\mathbb{P}^{X_{1:k}}$. For each of the $(T-k)$ transitions, there are m^k possible pasts of length k. For each of those pasts we have to define $(m-1)$ transition probabilities in order to fully determine the process.

k	m	T	$\gamma(k,m,T)$
0	2	10	10
0	3	10	20
0	4	10	30
0	4	100	300
0	10	100	900
1	2	10	19
1	3	10	56
1	4	10	111
1	4	100	1,191
1	10	100	8,919
2	2	10	35
2	3	10	152
2	4	10	399
2	4	100	4,719
2	10	100	88,299
5	2	10	191
5	3	10	2,672
5	4	10	16,383
5	4	100	292,863
5	10	100	85,599,999

Table 3.1: The model complexity $\gamma(k,m,T)$ of time-inhomogeneous, time-discrete Markov processes of different orders k, different lengths T and state spaces of different cardinality m

Bühlmann et al. (1999) stated, within a time-homogeneous setting, that "[t]here are no models in between [...]". This remains true in the current, time-inhomogeneous setting. For example, it is impossible to fit a time-inhomogeneous, time-discrete Markov process with 50 model parameters to data on a state space of cardinality $m = 3$ and length $T = 10$. In fact we can only fit a model with 30 parameters or with 111 parameters and nothing in between: "The class of all [...] full [time-discrete] Markov [processes] is not structurally rich [...]", cf. Mächler and Bühlmann (2004). On the contrary the class of all tiVLMC is rich, there are

endless possibilities to calibrate the number of model parameters by alternating the context tree.
The second big advantage is that tiVLMC, as we have already discussed shortly in Chapter 1.1, avoid the curse of dimensionality: As seen in Table 3.1, the model complexity of a time-inhomogeneous, time-discrete Markov process increases exponentially with the order k. Thus fitting such processes of higher order leads to highly variable estimates and insane amounts of training data are required to calibrate higher-order models. On the other hand a tiVLMC can describe the same dependency structure as its embedding while usually having drastically less model parameters and thus being a more robust model.

Example 3.15. The tiVLMC of Example 3.11 has

$$3 + 3 + 4 \cdot 3 + (5 \cdot 3)(T - 3)$$

model parameters. 3 parameters are needed to describe the initial distribution $\mathbb{P}^{X_1}$ of X: the probabilities to start in a, i or i_+. The probability to start in d is the complementary event and we do not need to estimate it. Since $\tau_1 = \{e\}$, X_2 is not depending on X_1 and we again need 3 parameters to describe $\mathbb{P}^{X_2}$. Since $|\tau_2| = 4$, we have 4 states (a, i, i_+ and d) relevant for the second transition and thus need $4 \cdot 3$ more model parameters. For each of the $(T - 3)$ transitions after the first two, we need $5 \cdot 3$ parameters, since we have $|\tau_t| = 5$ states (a, i, (i_+, w) with $w \neq i_+$, (i_+, i_+) and d), $t \geq 3$.
The embedding time-inhomogeneous, time-discrete Markov process of order 2 needs

$$15 + 3 \cdot 4^2 \cdot (T - 2)$$

model parameters. For $T = 50$ this amounts to more than three times more parameters than the tiVLMC has itself, cf. the following Table 3.2.

T	tiVLMC	embedding	ratio
5	48	159	3.31
10	123	399	3.24
25	348	1,119	3.22
50	723	2,319	3.21
100	1,473	4,719	3.20
500	7,473	23,919	3.20

Table 3.2: The model complexity of the tiVLMC from Example 3.11 and its embedding time-inhomogeneous, time-discrete Markov process of order two and their ratio for varying lengths T

The ratios in Table 3.2 are the parameters of the embedding time-inhomogeneous, time-discrete Markov process divided by the parameters of the tiVLMC, rounded to two decimal digits.
The more the lengths of the different contexts of a tiVLMC vary, the bigger the ratio is when model complexity is compared to their embedding.

4 Inferring tiVLMC

This chapter is devoted to the model fitting procedure in a data setting where censoring is absent. The more general setting with censored data is considered in the following Chapter 5.
First we are going to present tools needed to generalize the fitting algorithm Context, cf. Algorithm 1.11. We are going to continue by constructing the above mentioned generalization. A tiVLMC-version of Theorem 1.12 will be proven, stating that the context tree estimator output by the generalized algorithm is consistent. The major step of the Context algorithm is the pruning decision performed in the second algorithmic step. Here a measure is required that evaluates whether a leaf that is under consideration for pruning in fact should or should not be pruned. Conventionally the Kullback-Leibler divergence is chosen as the measure and this is how we presented Context in the introductory Chapter 1.2. Reasons for this convention are going to be explained within the following. However we are also going to consider other possible measure choices like the L^1-norm.

4.1 Pruning with the Kullback-Leibler divergence

First we formally introduce the Kullback-Leibler divergence as a measure of the distance between two distributions.

Definition 4.1. Let P and Q be two discrete probability distributions on Ω. The Kullback-Leibler divergence between P and Q is defined as

$$\mathrm{D}\left[P||Q\right] := \sum_{x\in\Omega} P(x)\log\frac{P(x)}{Q(x)},$$

where $P(x)\log\frac{P(x)}{Q(x)} := 0$ for $P(x) = 0$ and $P(x)\log\frac{P(x)}{0} := \infty$ for $P(x) > 0$.

The Kullback-Leibler divergence was introduced by and in Kullback and Leibler (1951). For a function $f : \Omega \to R$, where Ω is a set of finite cardinality, we denote the L^1-norm of f by

$$\|f\|_1 := \sum_{x\in\Omega} |f(x)|.$$

The L^1-distance between two discrete distributions can be rewritten:

Proposition 4.2 (Cover and Thomas, 1991)**.** *Let P and Q be two discrete probability distributions on Ω, where Ω is of finite cardinality. For*

$$A := \{x \in \Omega : P(x) > Q(x)\}$$

it holds that

$$\|P - Q\|_1 = 2\,(P(A) - Q(A)).$$

This proposition is proven in Cover and Thomas (1991, page 299). Using Proposition 4.2 one can obtain the following lower bound for the Kullback-Leibler divergence.

Proposition 4.3 (Cover and Thomas, 1991)**.** *Let P and Q be two discrete probability distributions on Ω, where Ω is of finite cardinality. Then*

$$\mathrm{D}\,[P||Q] \geq \frac{1}{2\log 2}\,\|P - Q\|_1^2\,.$$

This is Lemma 12.6.1 in Cover and Thomas (1991, page 300). By replacing the nominator $2\log 2$ by 2 one obtains a less strict inequality which is known as Pinsker's inequality, cf. Pinsker (1973). Combining Propositions 4.2 and 4.3 yields:

Corollary 4.4. *Let P and Q be two discrete probability distributions on Ω, where Ω is of finite cardinality. For*

$$A := \{x \in \Omega : P(x) > Q(x)\}$$

it holds that

$$\mathrm{D}\,[P||Q] \geq (P(A) - Q(A))^2.$$

Using above statements we can introduce the time-sensitive pendant of the original pruning measure used by Rissanen (1983). This tool plays the most important role in the pruning decision of the (to-be constructed) fitting Algorithm 4.7.

Definition 4.5. For $1 \leq t \leq T$, $wu \in M^{1:t}$ with $u \in M$ and $N_t(w) > 0$, we define the time-sensitive pruning measure

$$\Delta_t(wu) := N_t(wu)\mathrm{D}\left[\hat{P}_t(\cdot \mid wu)||\hat{P}_t(\cdot \mid w)\right].$$

We next establish a relation to compare the size of two (terminal node context or context) trees. This formalization is useful to propose a stopping criterion of Algorithm 4.7 that we present afterwards.

Definition 4.6. A set $\tau \subset M^{0:\infty}$ is said to be smaller or equally sized than another set $\mathcal{T} \subset M^{0:\infty}$, written $\tau \preccurlyeq \mathcal{T}$, if

$$w \in \tau \Rightarrow \left(\exists u \in M^{0:\infty} : wu \in \mathcal{T}\right).$$

We call τ and $\mathcal{T}$ equally sized, if $\tau \preccurlyeq \mathcal{T}$ and $\mathcal{T} \preccurlyeq \tau$ hold simultaneously.

Note that $\subseteq$ is not equivalent to $\preccurlyeq$. While $\{1\} \not\subseteq \{(1,1)\}$, it clearly holds that $\{1\} \preccurlyeq \{(1,1)\}$.
Now we present the algorithm to estimate the context function c of a tiVLMC $X \in \mathcal{P}^{0:T-1}$. As mentioned before this algorithm is a modified version of Context, cf. Algorithm 1.11. In contrast to the original algorithm it is able to meet the special requirements that come up in insurance applications (for uncensored data) and were discussed in Chapter 1.3. The case of censored data is dealt with separately in Chapter 5.

Algorithm 4.7.
Input: $X_{1:T}^1 = x_{1:T}^1, ..., X_{1:T}^n = x_{1:T}^n$, *a cutoff* $C > 0$ *and an integer* $n_{\min} \in \mathbb{N}$

For $t = 1, ..., T-1$ *do:*

Step 1: Tree growing

Construct the maximal terminal node context tree $\tau_{\max}^t$ *such that*

$$w \in \tau_{\max}^t \Rightarrow N_t(w) \geq n_{\min}$$

and

$$\forall \tau_t^t \text{ with } w \in \tau_t^t \Rightarrow N_t(w) \geq n_{\min} \text{ it holds that } \tau_t^t \preccurlyeq \tau_{\max}^t.$$

Set $\tau_{(0)}^t := \tau_{\max}^t$.

Step 2: Leaf pruning
Examine every element $wu \in \tau^t_{(0)}$, $u \in M$, *as follows:*
Prune wu *down to* w *if*

$$\Delta_t(wu) < C\sqrt{n \log n},$$

else do nothing.
This yields a smaller or equally sized tree $\tau_{(1)} \preccurlyeq \tau^t_{(0)}$.

Step 3: Stopping criterion
Repeat Step 2 with $\tau^t_{(i)}$ *instead of* $\tau^t_{(i-1)}$ *until* $\tau_{(i+1)}$ *and* $\tau^t_{(i)}$ *are equally sized. Denote this maximally pruned context tree* $\hat{\tau}_t$.

Output: $\hat{\tau} := (\hat{\tau}_t)_{1 \le t \le T-1}$

The basic idea of Algorithm 4.7 is the following: During every pruning decision we compare two possible context functions $\hat{c}$ and $\hat{c}'$, where $\hat{c}'$ is obtained by pruning the terminal node wu of $\hat{c}_t$ down to w. Thus $\hat{c}$ and $\hat{c}'$ only differ at time t and only at the terminal node wu of $\hat{c}_t$, where $\hat{c}'_t$ takes the value w. Now $\Delta_t(wu)$ measures whether the additional information contained in the longer context wu is worth keeping (so $\hat{c}$ is the better fitting context function) or should be dropped ($\hat{c}'$ is the better fitting context function). In (3.1) we have seen that the likelihood of observing $x_{1:T}$ is

$$\mathbb{P}\left(X_{1:T} = x_{1:T}\right) = \mathbb{P}(X_1 = x_1) \cdot \prod_{t=1}^{T-1} P_t\left(x_{t+1} \mid c_t(x_{1:t})\right),$$

if c is the true context function of the tiVLMC. Thus the estimated likelihood (conditioned on the first state) for observing $x^1_{1:T}, ..., x^n_{1:T}$ using the context function $\hat{c}$ is

$$L\left(\hat{c}\right) = \prod_{i=1}^{n} \prod_{t=1}^{T-1} \hat{P}_t\left(x^i_{t+1} \,\middle|\, \hat{c}_t\left(x^i_{1:t}\right)\right); \tag{4.1}$$

here we used that $X^1, ..., X^n$ are independent. Since $\hat{c}_k = \hat{c}'_k$ for every $k \neq t$, it holds that the log-ratio between the two likelihoods is

$$\log \frac{L\left(\hat{c}\right)}{L\left(\hat{c}'\right)} = \log \left(\frac{\prod_{i=1}^{n} \hat{P}_t\left(x^i_{t+1} \,\middle|\, \hat{c}_t\left(x^i_{1:t}\right)\right)}{\prod_{i=1}^{n} \hat{P}_t\left(x^i_{t+1} \mid \hat{c}'_t\left(x^i_{1:t}\right)\right)} \right).$$

Now if $\hat{c}_t\left(x_{1:T}^i\right) = wu$, it holds that $\hat{c}'_t\left(x_{1:T}^i\right) = w$. If $\hat{c}_t\left(x_{1:T}^i\right) \neq wu$, both contexts are equal, i.e. $\hat{c}_t\left(x_{1:T}^i\right) = \hat{c}'_t\left(x_{1:T}^i\right)$ in that case. Thus the only factors that remain in the fraction are those with $x_{t-|wu|+1:t}^i = wu$ and hence we get

$$\begin{aligned}
\log\left(\frac{\prod_{i=1}^{n}\hat{P}_t\left(x_{t+1}^i \,\middle|\, \hat{c}_t\left(x_{1:t}^i\right)\right)}{\prod_{i=1}^{n}\hat{P}_t\left(x_{t+1}^i \mid \hat{c}'_t\left(x_{1:t}^i\right)\right)}\right) &= \log\left(\frac{\prod_{x\in M}\hat{P}_t\left(x \mid wu\right)^{N_{t+1}(xwu)}}{\prod_{x\in M}\hat{P}_t\left(x \mid w\right)^{N_{t+1}(xwu)}}\right) \\
&= \sum_{x\in M} N_{t+1}(xwu)\log\left(\frac{\hat{P}_t\left(x \mid wu\right)}{\hat{P}_t\left(x \mid w\right)}\right) \\
&= N_t(wu)\sum_{x\in M}\hat{P}_t(x \mid wu)\log\left(\frac{\hat{P}_t\left(x \mid wu\right)}{\hat{P}_t\left(x \mid w\right)}\right) \\
&= N_t(wu)\mathrm{D}\left[\hat{P}_t(\cdot \mid wu)||\hat{P}_t(\cdot \mid w)\right] \\
&= \Delta_t(wu).
\end{aligned}$$

Therefore the time-sensitive pruning measure is nothing else than a likelihood ratio test statistic. The null hypothesis is $H_0 : c = \hat{c}$ (no pruning), the alternative $H_1 : c = \hat{c}'$ (pruning). This gives statistical reasoning for the choice of the time-sensitive pruning measure Δ_t and in particular for the choice of the Kullback-Leibler divergence D as its most important component.

As stated in Cover and Thomas (1991, page 18), the Kullback-Leibler divergence also plays a main role in information theory: The Kullback-Leibler divergence "[...] is a measure of the distance between two distributions. In statistics, it arises as an expected logarithm of the likelihood ratio. [...] $\mathrm{D}\left[P||Q\right]$ is a measure of the inefficiency of assuming that the distribution is Q when the true distribution is P. For example, if we knew the true distribution of the random variable, then we could construct a code with average description length $H(P)$. If, instead, we used the code for a distribution Q, we would need $H(P) + \mathrm{D}\left[P||Q\right]$ bits on the average to describe the random variable." Here

$$H(P) := -\sum_{x\in\Omega} P(x)\log P(x)$$

denotes the entropy of the discrete probability distribution P on Ω.

This gives another reasoning for the construction of the time-sensitive pruning measure, also cf. Begleiter et al. (2004, page 388) or Ron et al. (1996, page 118): By using the Kullback-Leibler divergence we measure the additional memory needed if we prune wu down to w, i.e. if we use the measure $\hat{P}_t(\cdot \mid w)$ with the shorter context instead of $\hat{P}_t(\cdot \mid wu)$.

In the second step of Algorithm 4.7 the informational benefit inherited by using wu instead of only w is compared to the threshold $C\sqrt{n \log n}$. Via the so-called cutoff value C the harshness of pruning, i.e. the harshness of the second step, can be calibrated: Larger values of C lead to more pruning. C is used as a tuning parameter and more details regarding the choice of C will be discussed in the later Chapter 6. The $\sqrt{n \log n}$-term arises as a needed technical growing factor in the proof of Theorem 4.14, i.e. it ensures the consistency of Algorithm 4.7.
The integer $n_{\min}$ lets the user control the growth of the maximal terminal node context tree $\tau_{\max}^{\mathsf{t}}$ in the first step of Algorithm 4.7. Low values for $n_{\min}$ generate really large trees $\tau_{\max}^{\mathsf{t}}$ and thus large amounts of memory (and also CPU processing power) are required to calculate the model fit. $n_{\min}$ lets the user limit the numerical effort and should be chosen as low as possible with the constraint that computations can be carried out on the given environment within the desired time frame.
The following theorem states that the estimator output by Algorithm 4.7 includes the true context tree with probability one if the number of observations diverges to infinity.

Theorem 4.8. *Let $X = (X_t)_{1 \le t \le T} \in \mathcal{P}^{0:T-1}$ with $T \ge 2$ be a tiVLMC with state space M of cardinality $2 \le m < \infty$ and (unknown) context tree τ. If $X_{1:T}^1 = x_{1:T}^1, ..., X_{1:T}^n = x_{1:T}^n$ are n independent observations of X and $\hat{\tau}$ is the output of Algorithm 4.7,*

$$\lim_{n \to \infty} \mathbb{P}(\tau_t \preccurlyeq \hat{\tau}_t) = 1$$

holds for all $1 \le t \le T - 1$.

Proof of Theorem 4.8. Fix t. The underestimation event is

$$U_n := \{\tau_t \preccurlyeq \hat{\tau}_t\}^C = \Big\{\exists w \in \hat{\tau}_t \, \exists u \in M^{1:t-|w|} : wu \in \tau_t, \, wu \notin \hat{\tau}_t\Big\}.$$

Define

$$D_n := \{\forall w \in \tau_t^{\mathsf{t}} : N_t(w) \ge \rho_n\},$$

where

$$\rho_n := \frac{bn}{2}$$

with

$$b := \min_{w \in \tau_t^{\mathsf{t}}} \mathbb{P}\left(X_{t-|w|+1:t} = w\right).$$

Then

$$U_n \subseteq (U_n \cap D_n) \cup D_n^C$$

and the σ-subadditivity of $\mathbb{P}$ implies

$$\mathbb{P}(U_n) \leq \mathbb{P}(U_n \cap D_n) + \mathbb{P}\left(D_n^C\right).$$

Since probabilities are non-negative this yields

$$0 \leq \lim_{n\to\infty} \mathbb{P}(U_n) \leq \lim_{n\to\infty} \mathbb{P}(U_n \cap D_n) + \lim_{n\to\infty} \mathbb{P}\left(D_n^C\right)$$

and hence the claim can be further split into two lemmas.

Lemma 4.9. *With above notations it holds that*

$$\lim_{n\to\infty} \mathbb{P}\left(D_n^C\right) = 0.$$

Lemma 4.10. *With above notations it holds that*

$$\lim_{n\to\infty} \mathbb{P}(U_n \cap D_n) = 0.$$

Using Lemma 4.9 and 4.10 we have

$$0 \leq \lim_{n\to\infty} \mathbb{P}(U_n) \leq 0 + 0,$$

and

$$\lim_{n\to\infty} \mathbb{P}(U_n) = 0$$

follows by the sandwich theorem A.4. □

Proof of Lemma 4.9. The complement of D_n is

$$D_n^C = \{\exists w \in \tau_t^{\mathsf{t}} : N_t(w) < \rho_n\}.$$

Using the σ-subadditivity of $\mathbb{P}$ and moving over to the largest summand we can bound the probability of the complement by

$$\begin{aligned}\mathbb{P}\left(D_n^C\right) &= \mathbb{P}\left(\exists w \in \tau_t^{\mathsf{t}} : N_t(w) < \rho_n\right) \\ &\leq \sum_{w \in \tau_t^{\mathsf{t}}} \mathbb{P}\left(N_t(w) < \rho_n\right) \\ &\leq |\tau_t^{\mathsf{t}}| \max_{w \in \tau_t^{\mathsf{t}}} \mathbb{P}\left(N_t(w) < \rho_n\right) \\ &\leq m^t \max_{w \in \tau_t^{\mathsf{t}}} \mathbb{P}\left(N_t(w) < \rho_n\right), \end{aligned} \tag{4.2}$$

where we have used that the cardinality $|\tau_t^{\mathsf{t}}|$ of the terminal node context tree τ_t^{t} at t is (loosely) upper bounded by the number of possible distinct sequences of values in M of length t, i.e.

$$|\tau_t^{\mathsf{t}}| \leq m^t.$$

Recall Definitions 3.1 and 3.3 and then note that

$$\text{domain}(c_t) = \text{tpsupp}_t(X) = \left\{x \in M^t : \mathbb{P}(X_{1:t} = x) > 0\right\}.$$

Thus we have

$$\forall w \in \tau_t \, \exists x \in \text{tpsupp}_t(X) : w = c_t(x)$$

and therefore

$$0 < \mathbb{P}(X_{1:t} = x) \leq \mathbb{P}(X_{t-|w|+1:t} = w)$$

for every $w \in \tau_t^{\mathsf{t}} \subseteq \tau_t$. Hence $b > 0$.
$N_t(w)$ is $\text{Bin}(n, \mathbb{P}(X_{t-|w|+1:t} = w))$-distributed. This yields

$$\mathbb{E}\left[N_t(w)\right] = n\mathbb{E}\left[\mathbb{1}\left(X_{t-|w|+1:t}^1 = w\right)\right] = n\mathbb{P}(X_{t-|w|+1:t} = w) \geq nb > 0.$$

With our choice of

$$\rho_n = \frac{bn}{2} > 0$$

we can continue to bind the probability in (4.2):

$$\begin{aligned}\mathbb{P}(N_t(w) < \rho_n) &= \mathbb{P}\left(N_t(w) - \mathbb{E}\left[N_t(w)\right] < \rho_n - \mathbb{E}\left[N_t(w)\right]\right) \\ &\leq \mathbb{P}\left(N_t(w) - \mathbb{E}\left[N_t(w)\right] < \rho_n - nb\right) \\ &= \mathbb{P}\left(N_t(w) - \mathbb{E}\left[N_t(w)\right] < -\frac{bn}{2}\right).\end{aligned}$$

The indicators $(N_t^i(w))_{i=1,\dots,n}$ are independent, since $X^1, \dots, X^n$ are, and bounded by the interval $[0,1]$. Since $-\frac{bn}{2} < 0$, we can apply Hoeffding's inequality A.7 and obtain

$$\begin{aligned}\mathbb{P}(N_t(w) < \rho_n) &\leq \mathbb{P}\left(N_t(w) - \mathbb{E}\left[N_t(w)\right] < -\frac{bn}{2}\right) \\ &\leq \exp\left(-2\frac{\left(-\frac{bn}{2}\right)^2}{n}\right) \\ &= \exp\left(-\frac{b^2}{2}n\right).\end{aligned}$$

Combining this with (4.2) we finally obtain

$$\begin{aligned}0 \leq \lim_{n\to\infty} \mathbb{P}\left(D_n^C\right) &\leq m^t \lim_{n\to\infty} \max_{w\in\tau_t^{\mathsf{t}}} \mathbb{P}\left(N_t(w) < \rho_n\right) \\ &\leq m^t \lim_{n\to\infty} \exp\left(-\frac{b^2}{2}n\right) \qquad (4.3) \\ &= m^t \cdot 0 \\ &= 0\end{aligned}$$

and thus

$$\lim_{n\to\infty} \mathbb{P}\left(D_n^C\right) = 0$$

by the sandwich theorem A.4. This completes the proof. □

Proof of Lemma 4.10. Without loss of generality underestimation can be restricted to the terminal nodes of τ_t: If an (internal) node of τ_t is missing, it must be that a terminal node of τ_t is also missing. Thus we can simplify the underestimation event:

$$U_n = \{\exists w \in \tau_t^{\mathsf{t}} : w \notin \hat{\tau}_t\} = \{\exists wu \in \tau_t^{\mathsf{t}},\, u \in M : wu \notin \hat{\tau}_t\}.$$

For a terminal node $wu \in \tau_t^{\mathsf{t}}$ two distinct reasons can cause underestimation:

1. wu was not even included in $\tau_{\max}^{\mathsf{t}}$; this happens if and only if $N_t(wu) < n_{\min}$.
2. wu was included in $\tau_{\max}^{\mathsf{t}}$ but falsely pruned down to w; this happens if and only if $N_t(wu) \geq n_{\min}$ and $\Delta_t(wu) < C\sqrt{n \log n}$.

Recall that we have chosen

$$\rho_n = \frac{bn}{2}$$

and that $b > 0$. Therefore

$$\rho_n \xrightarrow{n\to\infty} \infty$$

strictly monotonically. Thus there exists a threshold integer $n_\rho \in \mathbb{N}$ such that

$$\rho_n \geq n_{\min}$$

holds for all $n \geq n_\rho$. And hence for $n \geq n_\rho$

$$U_n \cap D_n = \left\{\exists wu \in \tau_t^{\mathtt{t}},\, u \in M : \Delta_t(wu) < C\sqrt{n \log n},\, N_t(wu) \geq \rho_n\right\}$$

holds, i.e. by cutting U_n with D_n we can restrict ourselves to the second reason for underestimation if n is large. By using the σ-subadditivity of $\mathbb{P}$ and plugging in Definition 4.5 of the time-sensitive pruning measure, we obtain the upper bound

$$\begin{aligned}
&\mathbb{P}(U_n \cap D_n) \\
&\leq \sum_{\substack{wu\in\tau_t^{\mathtt{t}}\\ u\in M}} \mathbb{P}\left(\Delta_t(wu) < C\sqrt{n\log n},\, N_t(wu) \geq \rho_n\right) \\
&= \sum_{\substack{wu\in\tau_t^{\mathtt{t}}\\ u\in M}} \mathbb{P}\left(\mathrm{D}\left[\hat{P}_t(\cdot \mid wu)||\hat{P}_t(\cdot \mid w)\right] < \frac{C\sqrt{n\log n}}{N_t(wu)},\, N_t(wu) \geq \rho_n\right) \\
&\leq \sum_{\substack{wu\in\tau_t^{\mathtt{t}}\\ u\in M}} \sum_{k=\lceil\rho_n\rceil}^{n} \mathbb{P}\left(\mathrm{D}\left[\hat{P}_t(\cdot \mid wu)||\hat{P}_t(\cdot \mid w)\right] < \frac{C\sqrt{n\log n}}{k},\, N_t(wu) = k\right) \\
&\leq \sum_{\substack{wu\in\tau_t^{\mathtt{t}}\\ u\in M}} \sum_{k=\lceil\rho_n\rceil}^{n} \sum_{j=k}^{n} \mathbb{P}\left(\mathrm{D}\left[\hat{P}_t(\cdot \mid wu)||\hat{P}_t(\cdot \mid w)\right] < \frac{C\sqrt{n\log n}}{k},\, N_t(wu) = k,\, N_t(w) = j\right)
\end{aligned} \tag{4.4}$$

for $n \geq n_\rho$. The range of the inner sum is due to

$$N_t(wu) = k \Rightarrow N_t(w) = \sum_{u\in M} N_t(wu) \geq k.$$

Since M is finite, we can apply Corollary 4.4 and lower bound the Kullback-Leibler divergence

$$\mathrm{D}\left[\hat{P}_t(\cdot \mid wu)||\hat{P}_t(\cdot \mid w)\right] \geq \left(\hat{P}_t(A \mid wu) - \hat{P}_t(A \mid w)\right)^2$$

for

$$A := \left\{x \in M : \hat{P}_t(x \mid wu) > \hat{P}_t(x \mid w)\right\}.$$

Therefore

$$\begin{aligned}
&\mathbb{P}\left(\mathrm{D}\left[\hat{P}_t(\cdot \mid wu)||\hat{P}_t(\cdot \mid w)\right] < \frac{C\sqrt{n\log n}}{k},\, N_t(wu)=k,\, N_t(w)=j\right)\\
\leq&\mathbb{P}\left(\hat{P}_t(A \mid wu) - \hat{P}_t(A \mid w) < \sqrt{\frac{C\sqrt{n\log n}}{k}},\, N_t(wu)=k,\, N_t(w)=j\right). \quad (4.5)
\end{aligned}$$

Now fix $wu \in \tau_t^{\mathsf{t}}$. Applying Lemma 3.12, there exist a state $x_{wu} \in M$ with $\varepsilon_{wu} > 0$ and a threshold integer n_{wu}, so that $x_{wu} \in A$ for $n \geq n_{wu}$. Using Lemma 3.13 with

$$\gamma = \sqrt{\frac{C\sqrt{n\log n}}{k}} \text{ and } \xi = \left(\frac{\varepsilon_{wu}}{2} - \gamma\right)^2$$

we can continue to bound (4.5):

$$\begin{aligned}
&\mathbb{P}\left(\hat{P}_t(A \mid wu) - \hat{P}_t(A \mid w) < \sqrt{\frac{C\sqrt{n\log n}}{k}},\, N_t(wu)=k, N_t(w)=j\right)\\
=&\mathbb{P}\left(\sum_{x\in A}\left(\hat{P}_t(x \mid wu) - \hat{P}_t(x \mid w)\right) < \sqrt{\frac{C\sqrt{n\log n}}{k}},\, N_t(wu)=k,\, N_t(w)=j\right)\\
\leq&\mathbb{P}\left(\hat{P}_t(x_{wu} \mid wu) - \hat{P}_t(x_{wu} \mid w) < \sqrt{\frac{C\sqrt{n\log n}}{k}},\, N_t(wu)=k,\, N_t(w)=j\right)\\
\leq&\mathbb{P}\left(\left|\hat{P}_t(x_{wu} \mid wu) - P_t(x_{wu} \mid wu)\right| > \sqrt{\xi},\, N_t(wu)=k\right)\\
&+\mathbb{P}\left(\left|\hat{P}_t(x_{wu} \mid w) - P_t(x_{wu} \mid w)\right| > \sqrt{\xi},\, N_t(w)=j\right) \quad (4.6)
\end{aligned}$$

for $n \geq n_{wu}$. Next let I_j denote the index of the j-th 1 in the 0-1-valued sequence $(N_t^i(wu))_{i=1,\ldots,n}$ and define $Y_j := \mathbb{1}\left(N_{t+1}^{I_j}(x_{wu}wu) = 1\right)$. Y_j indicates whether an observation that moved through wu transits into x_{wu} at $t+1$. Then

$$\begin{aligned}
&\mathbb{P}\left(\left|\hat{P}_t(x_{wu} \mid wu) - P_t(x_{wu} \mid wu)\right| > \sqrt{\xi},\, N_t(wu)=k\right)\\
=&\mathbb{P}\left(\left|\frac{1}{N_t(wu)}\sum_{j=1}^{N_t(wu)} Y_j - P_t(x_{wu} \mid wu)\right| > \sqrt{\xi},\, N_t(wu)=k\right)\\
\leq&\mathbb{P}\left(\left|\frac{1}{k}\sum_{j=1}^{k} Y_j - P_t(x_{wu} \mid wu)\right| > \sqrt{\xi}\right).
\end{aligned}$$

We want to apply Hoeffding's inequality A.6. The $(Y_j)_{j=1,\dots,k}$, being indicators, are all bounded by the interval $[0,1]$. They are independent, since $X^1,\dots,X^n$ are, and $\mathrm{Bin}\,(1, P_t(x_{wu} \mid wu))$-distributed. Therefore

$$\mathbb{E}\left[\frac{1}{k}\sum_{j=1}^{k} Y_j\right] = \mathbb{E}\left[Y_1\right] = P_t(x_{wu} \mid wu).$$

Thus

$$\begin{aligned}
&\mathbb{P}\left(\left|\hat{P}_t(x_{wu} \mid wu) - P_t(x_{wu} \mid wu)\right| > \sqrt{\xi},\, N_t(wu) = k\right)\\
\leq&\mathbb{P}\left(\left|\frac{1}{k}\sum_{j=1}^{k} Y_j - P_t(x_{wu} \mid wu)\right| > \sqrt{\xi}\right)\\
\leq&2e^{-2k\xi}
\end{aligned} \tag{4.7}$$

and analogously

$$\mathbb{P}\left(\left|\hat{P}_t(x_{wu} \mid w) - P_t(x_{wu} \mid w)\right| > \sqrt{\xi},\, N_t(w) = j\right) \leq 2e^{-2j\xi}$$

by Hoeffding's inequality A.6. Since k ranges from $\lceil \rho_n \rceil \geq \frac{bn}{2}$ to n we have

$$0 \leq \gamma = \sqrt{\frac{C\sqrt{n\log n}}{k}} \leq \sqrt{\frac{2C\sqrt{n\log n}}{bn}} \xrightarrow{n\to\infty} 0$$

and thus $\gamma \to 0$ by the sandwich theorem A.4. In consequence

$$\xi = \left(\frac{\varepsilon_{wu}}{2} - \gamma\right)^2 \xrightarrow{n\to\infty} \frac{\varepsilon_{wu}^2}{4} > 0.$$

Since $|\tau_t^{\mathsf{t}}| \leq m^t < \infty$ there exists a threshold integer n_ξ so that for all $wu \in \tau_t^{\mathsf{t}}$

$$\xi \geq \varepsilon := \frac{1}{5}\min_{wu\in\tau_t^{\mathsf{t}}} \varepsilon_{wu}^2 > 0$$

for all $n \geq n_\xi$. Thus for all $wu \in \tau_t^{\mathsf{t}}$ we have

$$\mathbb{P}\left(\left|\hat{P}_t(x_{wu} \mid wu) - P_t(x_{wu} \mid wu)\right| > \sqrt{\xi},\, N_t(wu) = k\right) \leq 2e^{-2k\varepsilon}$$

and

$$\mathbb{P}\left(\left|\hat{P}_t(x_{wu} \mid w) - P_t(x_{wu} \mid w)\right| > \sqrt{\xi},\, N_t(w) = j\right) \leq 2e^{-2j\varepsilon}$$

if $n \geq n_\xi$. Since j ranges from k to n, it also behaves asymptotically like n. By combining everything we finally obtain

$$\begin{aligned} 0 \leq \mathbb{P}(U_n \cap D_n) &\leq \sum_{\substack{wu \in \tau_t^{\mathsf{t}} \\ u \in M}} \sum_{k=\lceil \rho_n \rceil}^{n} \sum_{j=k}^{n} \left(2e^{-2k\varepsilon} + 2e^{-2j\varepsilon} \right) \\ &\leq |\tau_t^{\mathsf{t}}| \cdot n \cdot n \cdot 4 \cdot e^{-2\rho_n \varepsilon} \\ &\leq 4m^t n^2 e^{-bn\varepsilon} \end{aligned} \tag{4.8}$$

for $n \geq \max \left\{ n_\rho, n_\xi, \max\limits_{wu \in \tau_t^{\mathsf{t}}} n_{wu} \right\}$. Thus

$$0 \leq \lim_{n \to \infty} \mathbb{P}(U_n \cap D_n) = 0$$

and the claim is once again implied by the sandwich theorem A.4. This completes the proof. □

Within above proof an upper probability bound for the event of underestimation at a fixed time t has been developed to prove that the probability eventually converges to zero. Evaluating this bound empirically, i.e. replacing the probability measures by their empirical estimates, can give guidance on the risk of underestimation.

Corollary 4.11. *It holds that*

$$\mathbb{P}\left(U_n\right) \leq m^t \exp\left(-\frac{b^2}{n} \right) + 4m^t n^2 \exp\left(-bn\varepsilon \right)$$

for large n, where

$$b := \min_{w \in \tau_t^t} \mathbb{P}\left(X_{t-|w|+1:t} = w \right)$$

and

$$\varepsilon := \frac{1}{5} \min_{\substack{wu \in \tau_t^t \\ x \in M}} \left\{ P_t\left(x \mid wu\right) - P_t\left(x \mid w\right) : P_t\left(x \mid wu\right) - P_t\left(x \mid w\right) > 0 \right\}.$$

Proof of Corollary 4.11. As proven in (4.3) it holds that

$$\mathbb{P}\left(D_n^C \right) \leq m^t \exp\left(-\frac{b^2}{n} \right)$$

for large n. As proven in (4.8) we have that

$$\mathbb{P}\left(U_n \cap D_n\right) \leq 4m^t n^2 \exp\left(-bn\varepsilon'\right)$$

for

$$\varepsilon' := \frac{1}{5} \min_{wu \in \tau_t^{\mathrm{t}}} \varepsilon_{wu}^2$$

and n large. And since

$$\begin{aligned} 0 < \varepsilon &= \frac{1}{5} \min_{\substack{wu \in \tau_t^{\mathrm{t}} \\ x \in M}} \left\{P_t\left(x \mid wu\right) - P_t\left(x \mid w\right) : P_t\left(x \mid wu\right) - P_t\left(x \mid w\right) > 0\right\} \\ &\leq \frac{1}{5}\left(P_t\left(x_{wu} \mid wu\right) - P_t\left(x_{wu} \mid w\right)\right) \\ &\leq \frac{1}{5} \min_{wu \in \tau_t^{\mathrm{t}}} \varepsilon_{wu}^2 \\ &= \varepsilon', \end{aligned}$$

the claim

$$\mathbb{P}(U_n) \leq \mathbb{P}\left(D_n^C\right) + \mathbb{P}\left(U_n \cap D_n\right) \leq m^t \exp\left(-\frac{b^2}{n}\right) + 4m^t n^2 \exp\left(-bn\varepsilon\right)$$

is implied for large n. □

The next theorem states that the estimator output by Algorithm 4.7 is included in the true context tree with probability one if the number of observations diverges to infinity.

Theorem 4.12. *Let $X = (X_t)_{1 \leq t \leq T} \in \mathcal{P}^{0:T-1}$ with $T \geq 2$ be a tiVLMC with state space M of cardinality $2 \leq m < \infty$ and (unknown) context tree τ. If $X_{1:T}^1 = x_{1:T}^1, ..., X_{1:T}^n = x_{1:T}^n$ are n independent observations of X and $\hat{\tau}$ is the output of Algorithm 4.7,*

$$\lim_{n \to \infty} \mathbb{P}(\hat{\tau}_t \preccurlyeq \tau_t) = 1$$

holds for all $1 \leq t \leq T-1$.

In the proof of Theorem 4.12 we will apply an upper bound for the Kullback-Leibler divergence D. The bound is a generalization of Lemma A.5 by Dragomir et al. (2000).

Lemma 4.13. *Let P and Q be two discrete probability distributions on Ω with $Q(x) = 0 \Rightarrow P(x) = 0$ for all $x \in \Omega$. Then*

$$\mathrm{D}\left[P||Q\right] \leq \left(\sum_{x\in\Omega} \frac{P^2(x)}{Q(x)}\right) - 1,$$

where we set $\frac{0}{0} := 0$.

Proof of Lemma 4.13. It is well-known that for every real-valued differentiable function f that is defined on an interval $I \subset \mathbb{R}$ and that is strictly convex,

$$f(b) - f(a) \geq f'(a)(b - a)$$

holds for all $a, b \in I$. Choose $f(x) = -\log(x)$ and $I = (0, \infty)$. Then

$$\log a - \log b \geq \frac{1}{a}(a - b)$$

for all $a, b > 0$. Multiplying with $-b$ yields

$$b \log b - b \log a \leq \frac{b}{a}(b - a)$$

and this simplifies to

$$b \log \frac{b}{a} \leq \frac{b^2}{a} - b.$$

Since $Q(x) = 0 \Rightarrow P(x) = 0$ implies

$$P(x) > 0 \Rightarrow Q(x) > 0$$

by contraposition, we therefore get

$$P(x) \log \frac{P(x)}{Q(x)} \leq \frac{P(x)^2}{Q(x)} - P(x) \tag{4.9}$$

for all $x \in \Omega$ with $P(x) > 0$. Now compute

$$\begin{aligned}
\mathrm{D}\left[P||Q\right] &= \sum_{x\in\Omega} P(x) \log \frac{P(x)}{Q(x)} \\
&= \sum_{\substack{x\in\Omega: \\ P(x)>0}} P(x) \log \frac{P(x)}{Q(x)} + \sum_{\substack{x\in\Omega: \\ P(x)=0}} P(x) \log \frac{P(x)}{Q(x)}
\end{aligned}$$

$$= \sum_{\substack{x\in\Omega:\\ P(x)>0}} P(x)\log\frac{P(x)}{Q(x)} + \sum_{\substack{x\in\Omega:\\ P(x)=0}} 0\log\frac{0}{Q(x)}$$
$$= \sum_{\substack{x\in\Omega:\\ P(x)>0}} P(x)\log\frac{P(x)}{Q(x)}.$$

Using (4.9) yields

$$\begin{aligned}\mathrm{D}\left[P||Q\right] &= \sum_{\substack{x\in\Omega:\\ P(x)>0}} P(x)\log\frac{P(x)}{Q(x)}\\ &\leq \sum_{\substack{x\in\Omega:\\ P(x)>0}} \left(\frac{P(x)^2}{Q(x)} - P(x)\right)\\ &= \sum_{\substack{x\in\Omega:\\ P(x)>0}} \frac{P(x)^2}{Q(x)} - \sum_{\substack{x\in\Omega:\\ P(x)>0}} P(x)\\ &= \left(\sum_{\substack{x\in\Omega:\\ P(x)>0}} \frac{P(x)^2}{Q(x)}\right) - 1.\end{aligned}$$

Now we can just add the summands that are zero without changing anything and obtain

$$\mathrm{D}\left[P||Q\right] \leq \left(\sum_{x\in\Omega} \frac{P^2(x)}{Q(x)}\right) - 1.$$

□

Proof of Theorem 4.12. Fix t. The overestimation event is

$$O_n := \left\{\exists w \in \tau_t\, \exists u \in M^{1:t-|w|} : wu \notin \tau_t,\, wu \in \hat{\tau}_t\right\}.$$

This simplifies to

$$O_n = \{\exists w \in \tau_t^{\mathsf{t}}\, \exists u \in M : wu \in \hat{\tau}_t\}.$$

By expressing the event in terms of the time-sensitive pruning measure we get

$$\mathbb{P}(O_n) \leq \sum_{w\in\tau_t^{\mathsf{t}}} \sum_{u\in M} \mathbb{P}\left(\Delta_t(wu) \geq C\sqrt{n\log n},\, N_t(wu) \geq n_{\min}\right), \tag{4.10}$$

since $N_t(wu) \geq n_{\min}$ has to hold for wu to be added to the initial tree and $\Delta_t(wu) \geq C\sqrt{n \log n}$ has to hold so that wu does not get pruned down to w. M is finite and it holds that

$$\hat{P}_t(x \mid w) = 0 \Rightarrow \hat{P}_t(x \mid wu) = 0.$$

Thus we can apply Lemma 4.13 to obtain

$$\begin{aligned}
\Delta_t(wu) &= \mathrm{D}\left[\hat{P}_t(\cdot \mid wu) || \hat{P}_t(\cdot \mid w)\right] N_t(wu) \\
&\leq \left(\left(\sum_{x \in M} \frac{\hat{P}_t(x \mid wu)^2}{\hat{P}_t(x \mid w)}\right) - 1\right) N_t(wu) \\
&= \left(\sum_{x \in M} \frac{\hat{P}_t(x \mid wu)^2}{\hat{P}_t(x \mid w)} N_t(wu)\right) - N_t(wu) \\
&= \left(\sum_{x \in M} \frac{\hat{P}_t(x \mid wu)}{\hat{P}_t(x \mid w)} N_{t+1}(xwu)\right) - N_t(wu). \qquad (4.11)
\end{aligned}$$

Here we used that by Proposition 2.11

$$\hat{P}_t(x \mid wu) N_t(wu) = \frac{N_{t+1}(xwu)}{N_t(wu)} N_t(wu) = N_{t+1}(xwu).$$

Note that

$$N_t(wu) = \sum_{x \in M} N_{t+1}(xwu)$$

and thereby

$$\begin{aligned}
\Delta_t(wu) &\leq \left(\sum_{x \in M} \frac{\hat{P}_t(x \mid wu)}{\hat{P}_t(x \mid w)} N_{t+1}(xwu)\right) - N_t(wu) \\
&= \left(\sum_{x \in M} \frac{\hat{P}_t(x \mid wu)}{\hat{P}_t(x \mid w)} N_{t+1}(xwu)\right) - \sum_{x \in M} N_{t+1}(xwu) \\
&= \sum_{x \in M} \left(\frac{\hat{P}_t(x \mid wu)}{\hat{P}_t(x \mid w)} - 1\right) N_{t+1}(xwu). \qquad (4.12)
\end{aligned}$$

For arbitrary $1 \leq v \leq T$ and arbitrary $z \in M^{1:v}$ the indicators

$$\left(N_v^i(z)\right)_{i=1,\ldots,n}$$

are $\mathrm{Bin}(1, P_v(z))$-distributed, where $P_v(z) := \mathbb{P}\left(X_{v-|z|+1:v} = z\right)$, and independent, since $X^1, ..., X^n$ are independent. Therefore

$$\left(\frac{N_v^i(z) - P_v(z)}{\sqrt{P_v(z)\left(1 - P_v(z)\right)}} \right)_{i=1,\dots,n}$$

are also independent random variables with

$$\mathbb{E}\left[\frac{N_v^i(z) - P_v(z)}{\sqrt{P_v(z)\left(1 - P_v(z)\right)}} \right] = 0$$

and

$$\mathrm{Var}\left[\frac{N_v^i(z) - P_v(z)}{\sqrt{P_v(z)\left(1 - P_v(z)\right)}} \right] = 1.$$

Thus the law of the iterated logarithm A.8 implies

$$\limsup_{n\to\infty} \frac{\sum_{i=1}^{n} \frac{N_v^i(z) - P_v(z)}{\sqrt{P_v(z)(1-P_v(z))}}}{\sqrt{2n \log\log n}} = \limsup_{n\to\infty} \frac{N_v(z) - nP_v(z)}{\sqrt{P_v(z)\left(1 - P_v(z)\right)}\sqrt{2n \log\log n}} = 1 \quad (4.13)$$

$\mathbb{P}$-a.s. We are interested in the asymptotic behaviour of a summand of (4.12). The basic idea is that, if those summands grow slower than $C\sqrt{n \log n}$, the probabilities (4.10) of overestimation converge to zero as n diverges to infinity. Let $r := \mathcal{O}\left(\sqrt{n \log\log n}\right)$. We here use the Bachmann-Landau notation, cf. Definition A.9. By (4.13) we can write

$$\begin{aligned} N_{t+1}(xwu) &= nP_{t+1}(xwu) + r, \\ N_{t+1}(xw) &= nP_{t+1}(xw) + r, \\ N_t(wu) &= nP_t(wu) + r, \\ N_t(w) &= nP_t(w) + r \end{aligned}$$

and start to compute:

$$\begin{aligned} \frac{\hat{P}_t(x \mid wu)}{\hat{P}_t(x \mid w)} - 1 =& \frac{N_{t+1}(xwu)N_t(w)}{N_t(wu)N_{t+1}(xw)} - 1 \\ =& \frac{(nP_{t+1}(xwu) + r)(nP_t(w) + r)}{(nP_t(wu) + r)(nP_{t+1}(xw) + r)} - 1 \\ =& \frac{n^2 P_{t+1}(xwu)P_t(w) + nr(P_{t+1}(xwu) + P_t(w)) + r^2}{n^2 P_t(wu)P_{t+1}(xw) + nr(P_t(wu) + P_{t+1}(xw)) + r^2} - 1. \end{aligned}$$

Next we expand the one and then join the two fractions. This yields

$$\frac{\hat{P}_t(x \mid wu)}{\hat{P}_t(x \mid w)} - 1 = \frac{nr(P_{t+1}(xwu) + P_t(w) - P_{t+1}(xw) - P_t(wu)) + r^2}{n^2 P_t(wu)P_{t+1}(xw) + nr(P_t(wu) + P_{t+1}(xw)) + r^2}. \quad (4.14)$$

Note that we have left out the quadratic term

$$n^2(P_{t+1}(xwu)P_t(w) - P_{t+1}(xw)P_t(wu))$$

in the nominator: Since $w \in \tau_t^{\mathsf{t}}$, it holds that

$$P_t(x \mid wu) = \frac{P_{t+1}(xwu)}{P_t(wu)} = \frac{P_{t+1}(xw)}{P_t(w)} = P_t(x \mid w)$$

and therefore

$$P_{t+1}(xwu)P_t(w) = P_{t+1}(xw)P_t(wu).$$

Thus the left out term is equal to zero. Here the information that we are dealing with an overestimation event comes to play. Next

$$\begin{aligned}
&\left(\frac{\hat{P}_t(x \mid wu)}{\hat{P}_t(x \mid w)} - 1\right) N_{t+1}(xwu) \\
=&\frac{nr(P_{t+1}(xwu) + P_t(w) - P_{t+1}(xw) - P_t(wu)) + r^2}{n^2 P_t(wu)P_{t+1}(xw) + nr(P_t(wu) + P_{t+1}(xw)) + r^2}(nP_{t+1}(xwu) + r) \\
=&\frac{n^2 r(P_{t+1}(xwu)^2 + P_t(w)P_{t+1}(xwu) - P_{t+1}(xw)P_{t+1}(xwu) - P_t(wu)P_{t+1}(xwu))}{n^2 P_t(wu)P_{t+1}(xw) + nr(P_t(wu) + P_{t+1}(xw)) + r^2} \\
&+ \frac{nr^2(2P_{t+1}(xwu) + P_t(w) - P_{t+1}(xw) - P_t(wu)) + r^3}{n^2 P_t(wu)P_{t+1}(xw) + nr(P_t(wu) + P_{t+1}(xw)) + r^2} \\
&+ \frac{r^3}{n^2 P_t(wu)P_{t+1}(xw) + nr(P_t(wu) + P_{t+1}(xw)) + r^2}.
\end{aligned}$$

Reducing the first fraction by n^2 and the second by n yields

$$\begin{aligned}
&\left(\frac{\hat{P}_t(x \mid wu)}{\hat{P}_t(x \mid w)} - 1\right) N_{t+1}(xwu) \\
=&\frac{r(P_{t+1}(xwu)^2 + P_t(w)P_{t+1}(xwu) - P_{t+1}(xw)P_{t+1}(xwu) - P_t(wu)P_{t+1}(xwu))}{P_t(wu)P_{t+1}(xw) + \frac{r}{n}(P_t(wu) + P_{t+1}(xw)) + \frac{r^2}{n^2}} \\
&+ \frac{r^2(2P_{t+1}(xwu) + P_t(w) - P_{t+1}(xw) - P_t(wu)) + \frac{r^3}{n}}{nP_t(wu)P_{t+1}(xw) + r(P_t(wu) + P_{t+1}(xw)) + \frac{r^2}{n}} \\
&+ \frac{r^3}{n^2 P_t(wu)P_{t+1}(xw) + nr(P_t(wu) + P_{t+1}(xw)) + r^2}
\end{aligned}$$

and thus

$$\begin{aligned}
&\left(\frac{\hat{P}_t(x \mid wu)}{\hat{P}_t(x \mid w)} - 1\right) N_{t+1}(xwu) \\
=&\mathcal{O}\left(\sqrt{n \log\log n}\right) + \mathcal{O}\left(\frac{n \log\log n}{n}\right) + \mathcal{O}\left(\frac{(n \log\log n)^{\frac{3}{2}}}{n^2}\right) \\
=&\mathcal{O}\left(\sqrt{n \log\log n}\right).
\end{aligned}$$

Therefore

$$\Delta_t(wu) = \sum_{x \in M} \mathcal{O}\left(\sqrt{n \log\log n}\right) = \mathcal{O}\left(m\sqrt{n \log\log n}\right) = \mathcal{O}\left(\sqrt{n \log\log n}\right)$$

by (4.12), i.e.

$$\limsup_{n \to \infty} \frac{\Delta_t(wu)}{\sqrt{n \log\log n}} = \text{const.}$$

$\mathbb{P}$-a.s. Since the Kullback-Leibler divergence is non-negative, the time-sensitive pruning measure is non-negative and thus

$$\limsup_{n \to \infty} \left|\frac{\Delta_t(wu)}{\sqrt{n \log\log n}}\right| = \text{const.}$$

$\mathbb{P}$-a.s. Now

$$\begin{aligned}
0 \leq \limsup_{n \to \infty} \left|\frac{\Delta_t(wu)}{\sqrt{n \log n}}\right| &\leq \limsup_{n \to \infty} \left|\frac{\Delta_t(wu)}{\sqrt{n \log\log n}}\right| \cdot \limsup_{n \to \infty} \left|\sqrt{\frac{n \log\log n}{n \log n}}\right| \\
&= \text{const} \cdot 0 \\
&= 0
\end{aligned}$$

$\mathbb{P}$-a.s. In consequence

$$\frac{\Delta_t(wu)}{\sqrt{n \log n}} \xrightarrow[\mathbb{P}\text{-a.s.}]{n \to \infty} 0$$

by the sandwich theorem A.4. Since almost sure convergence implies convergence in probability, for all $\varepsilon > 0$

$$\mathbb{P}\left(\Delta_t(wu) > \varepsilon\sqrt{n \log n}\right) \xrightarrow{n \to \infty} 0$$

holds, in particular for $\varepsilon = \frac{C}{2} > 0$. Thus

$$0 \leq \lim_{n \to \infty} \mathbb{P}\left(\Delta_t(wu) \geq C\sqrt{n \log n}\right) \leq \lim_{n \to \infty} \mathbb{P}\left(\Delta_t(wu) > \frac{C}{2}\sqrt{n \log n}\right) = 0$$

and hence

$$\lim_{n\to\infty} \mathbb{P}\left(\Delta_t(wu) \geq C\sqrt{n \log n}\,, N_t(wu) \geq 2\right) = 0$$

is again implied by the sandwich theorem A.4. Combining this with the fact that the number $|\tau_t^{\mathsf{t}}||M| \leq m^{t+1} < \infty$ of summands in (4.10) is finite, implies

$$\lim_{n\to\infty} \mathbb{P}(O_n) = 0.$$

This completes the proof. □

Theorem 4.8 says that the probability of underestimating the true context tree converges to zero. Theorem 4.12 says that the probability of overestimating the true context tree also converges to zero. This implies that the estimator output by Algorithm 4.7 is consistent. It allows us to estimate unknown context trees of data generating tiVLMC and therefore it is a strong instrument for analysing dependency structures of time-discrete stochastic processes with finite, unordered state spaces.
Note that it is common practice in statistics (and its subcategory of machine learning) to split up a consistency claim into two subclaims, e.g. cf. Leonardi et al. (2010, pages 326-330) or Bühlmann et al. (1999, pages 500-508): Asymptotically neither overestimation nor underestimation can occur.

Corollary 4.14. *Let $X = (X_t)_{1\leq t\leq T} \in \mathcal{P}^{0:T-1}$ with $T \geq 2$ be a tiVLMC with state space M of cardinality $2 \leq m < \infty$ and (unknown) context tree τ. If $X_{1:T}^1 = x_{1:T}^1, ..., X_{1:T}^n = x_{1:T}^n$ are n independent observations of X and $\hat{\tau}$ is the output of Algorithm 4.7,*

$$\lim_{n\to\infty} \mathbb{P}(\hat{\tau}_t = \tau_t) = 1$$

holds for all $1 \leq t \leq T-1$.

Proof of Corollary 4.14. Fix t. Define the error event

$$E_n := \{\tau_t \neq \hat{\tau}_t\}.$$

Then the claim is equivalent to

$$\lim_{n\to\infty} \mathbb{P}(E_n) = 0.$$

The error event E_n can be decomposed into the two possible types of estimation errors, over- and underestimation:

$$E_n \subseteq U_n \cup O_n.$$

Thus, by using the σ-subadditivity of $\mathbb{P}$,

$$\mathbb{P}(E_n) \leq \mathbb{P}(U_n) + \mathbb{P}(O_n).$$

Since probabilities are non-negative, this implies

$$0 \leq \lim_{n\to\infty} \mathbb{P}(E_n) \leq \lim_{n\to\infty} \mathbb{P}(U_n) + \lim_{n\to\infty} \mathbb{P}(O_n).$$

Using Theorem 4.8 and Theorem 4.12 we have

$$0 \leq \lim_{n\to\infty} \mathbb{P}(E_n) \leq 0 + 0 = 0,$$

and

$$\lim_{n\to\infty} \mathbb{P}(E_n) = 0$$

follows as a direct implication of the sandwich theorem A.4. □

Since the proof of the consistency of Algorithm 4.7, i.e. the proof of Corollary 4.14, has a pointwise structure in the time t, $1 \leq t \leq T-1$, it immediately follows that we could also allow for time-varying cutoff values $C_1, ..., C_{T-1} > 0$ instead of one cutoff $C > 0$ that is constant. The value of C determines the harshness of the pruning procedure performed in step two of the algorithm and therefore allowing the cutoff to depend on the time offers a possibility to equip the times with different weights. We will look at another possibility on how to do this in Chapter 11.2. But, since C is handled as a tuning parameter (the actual tuning procedure is discussed in Chapter 6) time-varying cutoff values would require to tune over a $(T-1)$-dimensional grid rather than a one-dimensional one. For large values of T this is computationally infeasible.

4.2 Pruning with the L^1-norm

The time-sensitive pruning measure defined in Definition 4.5 and used in Algorithm 4.7 does not need to be based on the Kullback-Leibler divergence. A

possible alternative is the L^1-norm.

Definition 4.15. For $1 \leq t \leq T$, $wu \in M^{1:t}$, with $u \in M$ and $N_t(w) > 0$, we define the L^1-norm based time-sensitive pruning measure

$$\Delta_t^{L^1}(wu) := N_t(wu)\|\hat{P}_t(\cdot \mid wu) - \hat{P}_t(\cdot \mid w)\|_1^2.$$

Proposition 4.3 proposed a relation between the Kullback-Leibler divergence and the L^1-norm. By this relation it holds that

$$\begin{aligned}\Delta_t(wu) = \mathrm{D}\left[\hat{P}_t(\cdot \mid wu)||\hat{P}_t(\cdot \mid w)\right] N_t(wu) &\geq \frac{1}{2}\|\hat{P}_t(\cdot \mid wu) - \hat{P}_t(\cdot \mid w)\|_1^2 N_t(wu) \\ &= \frac{1}{2}\Delta_t^{L^1}(wu).\end{aligned}$$

Therefore

$$2\Delta_t(wu) \geq \Delta_t^{L^1}(wu). \tag{4.15}$$

As a direct implication we obtain that the probability of overestimation also converges to zero, if one uses the L^1-norm based time-sensitive pruning measure $\Delta_t^{L^1}$ instead of Δ_t:

Theorem 4.16. *Theorem 4.12 remains true if Δ_t in Algorithm 4.7 is replaced by $\Delta_t^{L^1}$, i.e. the probability to overestimate the true context tree still vanishes.*

Proof of Theorem 4.16. Analogously to the proof of Theorem 4.12 we get

$$\mathbb{P}(O_n) \leq \sum_{w \in \tau_t^{\mathrm{t}}} \sum_{u \in M} \mathbb{P}\left(\Delta_t^{L^1}(wu) \geq C\sqrt{n \log n},\, N_t(wu) \geq n_{\min}\right).$$

Using the inequality (4.15) one obtains

$$\begin{aligned}&\mathbb{P}\left(\Delta_t^{L^1}(wu) \geq C\sqrt{n \log n},\, N_t(wu) \geq n_{\min}\right) \\ \leq &\mathbb{P}\left(\Delta_t(wu) \geq \frac{C}{2}\sqrt{n \log n},\, N_t(wu) \geq n_{\min}\right).\end{aligned}$$

Since $\frac{C}{2} > 0$ if $C > 0$,

$$\lim_{n \to \infty} \mathbb{P}(O_n) = 0$$

follows as in the proof of Theorem 4.12. □

In addition, as Theorem 4.8 states for the time-sensitive pruning measure, the event of underestimation also converges to zero if the L^1-norm based time-sensitive pruning measure is used:

Theorem 4.17. *Theorem 4.8 remains true if Δ_t in Algorithm 4.7 is replaced by $\Delta_t^{L^1}$, i.e. the probability to underestimate the true context tree still vanishes.*

Proof of Theorem 4.17. Theorem 4.8 was split into two Lemmas. Lemma 4.9 does not depend on the choice of the time-sensitive pruning measure. Hence, to prove the claimed convergence, it is sufficient to show that Lemma 4.10 remains true if Δ_t is replaced by $\Delta_t^{L^1}$. Because of the replacement the probability in (4.4) of the proof of Lemma 4.10 changes to

$$\mathbb{P}\left(\|\hat{P}_t(\cdot \mid wu) - \hat{P}_t(\cdot \mid w)\|_1^2 < \frac{C\sqrt{n \log n}}{k},\, N_t(wu) = k,\, N_t(w) = j\right).$$

Now note that

$$\begin{aligned}\|\hat{P}_t(\cdot \mid wu) - \hat{P}_t(\cdot \mid w)\|_1 &= \sum_{x \in M} \left|\hat{P}_t(x \mid wu) - \hat{P}_t(x \mid w)\right| \\ &\geq \sum_{x \in B} \left|\hat{P}_t(x \mid wu) - \hat{P}_t(x \mid w)\right|\end{aligned}$$

for all subsets $B \subseteq M$. Thus, for

$$A := \left\{x \in M : \hat{P}_t(x \mid wu) > \hat{P}_t(x \mid w)\right\},$$

we get

$$\begin{aligned}&\mathbb{P}\left(\|\hat{P}_t(\cdot \mid wu) - \hat{P}_t(\cdot \mid w)\|_1^2 < \frac{C\sqrt{n \log n}}{k},\, N_t(wu) = k,\, N_t(w) = j\right) \\ \leq &\mathbb{P}\left(\hat{P}_t(A \mid wu) - \hat{P}_t(A \mid w) < \sqrt{\frac{C\sqrt{n \log n}}{k}},\, N_t(wu) = k,\, N_t(w) = j\right).\end{aligned}$$

This expression is the same as (4.5) in the proof of Lemma 4.10 and thus Lemma 4.10 remains true if we exchange the time-sensitive pruning measure with $\Delta_t^{L^1}$. This completes the proof. □

Combining the two Theorems 4.16 and 4.17 (as in the proof of Corollary 4.14) implies the consistency of the estimator output by Algorithm 4.7, when $\Delta_t^{L^1}$ is used.

Corollary 4.18. *Corollary 4.14 remains true if Δ_t in Algorithm 4.7 is replaced by $\Delta_t^{L^1}$, i.e. the output estimator is still consistent.*

4.3 Pruning with an arbitrary norm

Since M is of finite cardinality m, the space of all real-valued functions $f : M \to \mathbb{R}$ defined on M is an m-dimensional vector space over the field $\mathbb{R}$, cf. e.g. Liesen and Mehrmann (2015, page 116). It is well-known that all norms defined on a vector space of finite dimension are equivalent, cf. Theorem A.10. Thus Corollary 4.18 implies the following general result.

Theorem 4.19. *Let $X = (X_t)_{1 \le t \le T} \in \mathcal{P}^{0:T-1}$ with $T \ge 2$ be a tiVLMC with state space M of cardinality $2 \le m < \infty$ and (unknown) context tree τ. If $X_{1:T}^1 = x_{1:T}^1, ..., X_{1:T}^n = x_{1:T}^n$ are n independent observations of X and $\hat{\tau}$ is the output of Algorithm 4.7, where $\Delta_t(wu)$ is replaced by*

$$\Delta_t^{\|\cdot\|}(wu) := N_t(wu) \left\| \hat{P}_t(\cdot \mid wu) - \hat{P}_t(\cdot \mid w) \right\|^2$$

and $\|\cdot\|$ is an arbitrary norm defined on the space of all functions $f : M \to \mathbb{R}$,

$$\lim_{n \to \infty} \mathbb{P}(\hat{\tau}_t = \tau_t) = 1$$

holds for all $1 \le t \le T - 1$.

Proof of Theorem 4.19. Since the space of all functions $f : M \to \mathbb{R}$ is of finite dimension, Theorem A.10 states that the norm $\|\cdot\|$ is equivalent to $\|\cdot\|_1$. Thus there exist constants $b > 0$ and $B > 0$ such that

$$b \left\| \hat{P}_t(\cdot \mid wu) - \hat{P}_t(\cdot \mid w) \right\|_1 \le \left\| \hat{P}_t(\cdot \mid wu) - \hat{P}_t(\cdot \mid w) \right\| \le B \left\| \hat{P}_t(\cdot \mid wu) - \hat{P}_t(\cdot \mid w) \right\|_1$$

holds for every possible sequence wu.
Analogously as in the proof of Theorem 4.16 we get that

$$\begin{aligned}\mathbb{P}(O_n) &\leq \sum_{w\in\tau_t^{\mathrm{t}}}\sum_{u\in M}\mathbb{P}\left(\Delta_t^{\|\cdot\|}(wu)\geq C\sqrt{n\log n},\, N_t(wu)\geq n_{\min}\right)\\ &\leq \sum_{w\in\tau_t^{\mathrm{t}}}\sum_{u\in M}\mathbb{P}\left(B^2\Delta_t^{L^1}(wu)\geq C\sqrt{n\log n},\, N_t(wu)\geq n_{\min}\right)\\ &\leq \sum_{w\in\tau_t^{\mathrm{t}}}\sum_{u\in M}\mathbb{P}\left(\Delta_t^{L^1}(wu)\geq \frac{C}{B^2}\sqrt{n\log n},\, N_t(wu)\geq n_{\min}\right)\end{aligned}$$

holds and since $\frac{C}{B^2} > 0$ for $B > 0$ and $C > 0$, this converges to zero.
Moreover it holds that

$$\begin{aligned}&\mathbb{P}\left(\|\hat{P}_t(\cdot\mid wu)-\hat{P}_t(\cdot\mid w)\|^2 < \frac{C\sqrt{n\log n}}{k},\, N_t(wu)=k,\, N_t(w)=j\right)\\ \leq&\mathbb{P}\left(b^2\|\hat{P}_t(\cdot\mid wu)-\hat{P}_t(\cdot\mid w)\|_1^2 < \frac{C\sqrt{n\log n}}{k},\, N_t(wu)=k,\, N_t(w)=j\right)\\ =&\mathbb{P}\left(\|\hat{P}_t(\cdot\mid wu)-\hat{P}_t(\cdot\mid w)\|_1^2 < \frac{C\sqrt{n\log n}}{b^2k},\, N_t(wu)=k,\, N_t(w)=j\right).\end{aligned}$$

Since $\frac{C}{b^2} > 0$ for $C > 0$ and $b > 0$, it follows, as it did in the proof of Theorem 4.17, that the probability of underestimation converges to zero as $n \to \infty$.

□

Before ending this chapter we want to raise the reader's awareness to the fact that Theorem 4.19 is an uncommon result. Here the finite dimensionality of the vector space implies its trueness. If one works in an infinite dimensional environment, there are many examples, e.g. from the mathematical discipline of robust statistics, where statements that are true for the L^1-norm are false for a different measure choice. The interested reader is referred to e.g. Rieder (2012) or Ruckdeschel and Rieder (2004).

5 Inferring tiVLMC from censored data

The data available to be used as input data for Algorithm 4.7 is often subject to censoring.

As a simple example assume that we observe a group of n policy holders who all bought the same contract. Each month we document the health status X_t^i at age t of policy holder number i for all policy holders $i = 1, ..., n$. The observation period starts at a fixed calendar time and ends at a later fixed calendar time. Since the policy holders do most likely not all sign the contract at exactly the same age, there will most likely exist ages t where X_t^i is only observed for certain, but not all i. As it is customary, we call the processes X^i, for which X_t^i is not observed, censored at t.

Censoring forces us to move away from the rectangular data setting, where n observations of length T are observed at the same t and therefore form a rectangular data matrix of size $n \times T$. We have to extend the theory of Chapter 4 to an environment where observations are allowed to be of different length and/or are observed at different t.

5.1 The censoring variables

Here we choose the modelling approach introduced by and in Phelan (1988): Censoring is represented by so-called "censoring processes" $J^1, ..., J^n$. Every J^i is a binary, discrete stochastic process, which indicates whether the i-th observation X^i is observable or whether it is not: If $J_t^i = 1$, the value of X_t^i is known, if $J_t^i = 0$, it is censored and thus unknown.

To generalize the fitting Algorithm 4.7 for tiVLMC developed in the uncensored setup, cf. the previous Chapter 4, we first need to update some definitions in order to integrate the censoring processes.

Definition 5.1. For $i = 1, ..., n$, $w \in M^{1:t}$ and $|w| \leq t \leq T$ let

$$\tilde{N}_t^i(w) := N_t^i(w) \cdot \mathbb{1}\left(J_{t-|w|+1:t}^i = (1, ..., 1)\right) \tag{5.1}$$

indicate whether X^i has moved through w till t and was observable, i.e. not censored. For $|w| \leq t \leq T - 1$ let

$$\tilde{N}_{t+}^i(w) := \tilde{N}_t^i(w) \cdot J_{t+1}^i \tag{5.2}$$

denote whether X^i has moved through w till t and was observable and additionally stays observable at $t+1$. Again, we denote the aggregations over i by

$$\tilde{N}_t(w) := \sum_{i=1}^{n} \tilde{N}_t^i(w)$$

and

$$\tilde{N}_{t+}(w) := \sum_{i=1}^{n} \tilde{N}_{t+}^i(w).$$

Definition 5.2. For $1 \leq t \leq T-1$, $x \in M$ and $w \in M^{1:t}$ with $\tilde{N}_{t+}(w) > 0$ we estimate the transition probability $P_t(x \mid w)$ by

$$\tilde{P}_t(x \mid w) := \frac{\tilde{N}_{t+1}(xw)}{\tilde{N}_{t+}(w)}.$$

To obtain the following statements, we will apply some assumptions on the censoring variables $J^1, ..., J^n$ and their dependence on the observations $X^1, ..., X^n$.

(H.1) The censoring variables $J^1, ..., J^n$ are identically distributed.

(H.2) The censoring variables $J^1, ..., J^n$ are independent.

(H.3) For $i = 1, ..., n$ the observation X^i and its censoring variable J^i are independent.

(H.4) For $i, j = 1, ..., n$ with $i \neq j$ the observation X^i and the censoring variable J^j are independent.

(H.5) For every $1 \leq t \leq T-1$ and every $w \in \text{tpsupp}_t(X)$

$$j_t(w) := \mathbb{P}\left(J^i_{t-|w|+1:t+1} = (1, ..., 1)\right) > 0,$$

$i = 1, ..., n$, holds. Note that $j_t(w)$ really only depends on $|w|$.

First note that this censored setting includes the less general uncensored setting discussed in Chapter 4: If $j_t(w) = 1$ for all combinations of t and w, the uncensored case is re-established.
Next note that one could summarize the assumptions (H.1), (H.2), (H.3) and (H.4). We choose not to do so since this enables us to give more precise references

in the following.
Lastly note that in the statistical literature such a censoring environment classifies as "missing completely at random", cf. e.g. Little and Rubin (2019) or Heitjan and Basu (1996).
If the number $N_t(w)$ of observations going through w till t is positive, it is implied that $\mathbb{P}(X_{t-|w|+1:t} = w) > 0$, i.e. $w \in \text{tpsupp}_t(X)$. If there is no censoring, we can estimate the transition probability $P_t(x \mid w)$ for every $x \in M$, cf. Proposition 2.11, if n is large enough. If $\tilde{N}_{t+}(w) = 0$ and $N_t(w) > 0$ hold simultaneously, $j_t(w) = 0$ has to hold and (H.5) is violated. In this case every observation is censored and naturally we cannot estimate a transition probability. Thus it is a natural restriction to assume (H.5), i.e. censoring is not allowed to be too heavy. This assumption guarantees, if the observation size n is large, that we get bias-free estimates of all non-zero transition probabilities and as will be shown afterwards these estimates also remain consistent.

Proposition 5.3. *Assume (H.1), (H.3) and (H.4). For $1 \leq t \leq T-1$, $x \in M$ and $w \in M^{1:t}$ with $\tilde{N}_{t+}(w) > 0$*

$$\mathbb{E}\left[\tilde{P}_t(x \mid w)\right] = P_t(x \mid w),$$

i.e. $\tilde{P}(x \mid w)$ is an unbiased estimator for $P_t(x \mid w)$.

Proof of Proposition 5.3. Since $\tilde{N}_{t+}(w) > 0$, it has to hold that $j_t(w) > 0$. Thus assuming $\tilde{N}_{t+}(w) > 0$ is even stronger than (H.5). Using (H.3) and (H.1) we can calculate the expectation of $\tilde{N}^i_{t+1}(xw)$:

$$\begin{aligned}
\mathbb{E}\left[\tilde{N}^i_{t+1}(xw)\right] &= \mathbb{E}\left[N^i_{t+1}(xw) \cdot \mathbb{1}\left(J^i_{t-|xw|+2:t+1} = (1, ..., 1)\right)\right] \\
&= \mathbb{E}\left[N^i_{t+1}(xw)\right] \cdot \mathbb{E}\left[\mathbb{1}\left(J^i_{t-|w|+1:t+1} = (1, ..., 1)\right)\right] \\
&= \mathbb{P}(X_{t-|xw|+2:t+1} = xw) \cdot j_t(w). \qquad (5.3)
\end{aligned}$$

Analogously

$$\mathbb{E}\left[\tilde{N}^i_{t+}(w)\right] = \mathbb{P}(X_{t-|w|+1:t} = w) \cdot j_t(w). \qquad (5.4)$$

Thus

$$\mathbb{E}\left[\tilde{N}^i_{t+1}(xw) \,\middle|\, \tilde{N}^i_{t+}(w) = 1\right] = P_{t+1}(x \mid w) \qquad (5.5)$$

and the claim

$$\mathbb{E}\left[\tilde{P}_t(x \mid w)\right] = P_t(x \mid w)$$

follows by copying the proof of Theorem 2.12. □

Proposition 5.4. *Assume (H.1)-(H.5). It holds that*

$$\max \left|\tilde{P}_t(x \mid w) - P_t(x \mid w)\right| \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} 0,$$

where the maximum runs over all well defined transition probabilities, i.e. over $1 \leq t \leq T-1$, $x \in M$ *and* $w \in M^{1:t}$ *with* $\mathbb{P}(X_{t-|w|+1:t} = w) > 0$.

Proof of Proposition 5.4. (H.1), (H.2) and (H.4) and the fact that the observations $X^1, ..., X^n$ are independent and identically distributed imply that the same is true for

$$\left(\tilde{N}^i_{t+1}(xw)\right)_{1\leq i\leq n}.$$

Thus by (H.3), (5.3) and the fact that indicators have bounded first absolute moments

$$\frac{\tilde{N}_{t+1}(xw)}{n} \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} \mathbb{P}(X_{t-|xw|+2:t+1} = xw) \cdot j_t(w)$$

is implied by the strong law of large numbers A.1. Analogously

$$\frac{\tilde{N}_{t+}(w)}{n} \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} \mathbb{P}(X_{t-|w|+1:t} = w) \cdot j_t(w).$$

Next the continuous mapping theorem A.11 implies

$$\frac{n}{\tilde{N}_{t+}(w)} \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} \frac{1}{\mathbb{P}(X_{t-|w|+1:t} = w) \cdot j_t(w)}.$$

Note that $j_t(w) > 0$ by (H.5) and $\mathbb{P}(X_{t-|w|+1:t} = w) > 0$ by assumption. Therefore

$$\tilde{P}_t(x \mid w) = \frac{\tilde{N}_{t+1}(xw)}{n}\frac{n}{\tilde{N}_{t+}(w)} \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} \frac{\mathbb{P}(X_{t-|xw|+2:t+1} = xw) \cdot j_t(w)}{\mathbb{P}(X_{t-|w|+1:t} = w) \cdot j_t(w)} = P_t(x \mid w).$$

Since $\mathscr{P}\left(M^T\right)$ consists of $2^{m^T} < \infty$ elements, cf. Proposition A.3, the range of the maximum is finite. This completes the proof. □

Hence we get counterparts to Theorem 2.9 and Theorem 2.12 in the presence of censoring and thus the defined estimator is a reasonable choice.

5.2 Pruning with the Kullback-Leibler divergence

By replacing the non-censored frequency counter N_t, cf. Definition 2.10, by its censored versions $\tilde{N}_t$ and $\tilde{N}_{t+}$, cf. Definition 5.1, we can define the censored time-sensitive pruning measure and elevate Algorithm 4.7 to the censored data setting.

Definition 5.5. For $1 \leq t \leq T$, $wu \in M^{1:t}$, where $u \in M$ and $\tilde{N}_{t+}(w) > 0$, we define the censored time-sensitive pruning measure

$$\tilde{\Delta}_t(wu) := \tilde{N}_{t+}(wu)\mathrm{D}\left[\tilde{P}_t(\cdot \mid wu)||\tilde{P}_t(\cdot \mid w)\right].$$

Algorithm 5.6.
Input: $X_{1:T}^1 = x_{1:T}^1, ..., X_{1:T}^n = x_{1:T}^n$, $J_{1:T}^1 = j_{1:T}^1, ..., J_{1:T}^n = j_{1:T}^n$, *a cutoff* $C > 0$ *and an integer* $n_{\min} \in \mathbb{N}$

For $t = 1, ..., T-1$ *do:*

Step 1: Tree growing
Construct the maximal terminal node context tree $\tau_{\max}^t$ *such that*

$$w \in \tau_{\max}^t \Rightarrow \tilde{N}_{t+}(w) \geq n_{\min}$$

and

$$\forall \tau_t^t \text{ with } w \in \tau_t^t \Rightarrow \tilde{N}_{t+}(w) \geq n_{\min} \text{ it holds that } \tau_t^t \preccurlyeq \tau_{\max}^t.$$

Set $\tau_{(0)}^t := \tau_{\max}^t$.

Step 2: Leaf pruning
Examine every element $wu \in \tau_{(0)}^t$, $u \in M$, *as follows:*
Prune wu *down to* w *if*

$$\tilde{\Delta}_t(wu) < C\sqrt{n \log n},$$

else do nothing.
This yields a smaller or equally sized tree $\tau_{(1)} \preccurlyeq \tau_{(0)}^t$.

Step 3: Stopping criterion
Repeat Step 2 with $\tau_{(i)}^t$ *instead of* $\tau_{(i-1)}^t$ *until* $\tau_{(i+1)}$ *and* $\tau_{(i)}^t$ *are equally*

sized. Denote this maximally pruned context tree $\tilde{\tau}_t$.

Output: $\tilde{\tau} := (\tilde{\tau}_t)_{1 \leq t \leq T-1}$

The probability to underestimate the true context tree still vanishes under censoring. We state and prove this in the next theorem.

Theorem 5.7. *Let* $X = (X_t)_{1 \leq t \leq T} \in \mathcal{P}^{0:T-1}$ *with* $T \geq 2$ *be a tiVLMC with state space* M *of cardinality* $2 \leq m < \infty$ *and (unknown) context tree* τ. *If* $X^1_{1:T} = x^1_{1:T}, ..., X^n_{1:T} = x^n_{1:T}$ *are* n *independent observations of* X, *censored by censoring variables* $J^1_{1:T} = j^1_{1:T}, ..., J^n_{1:T} = j^n_{1:T}$ *that satisfy (H.1)-(H.5), and* $\tilde{\tau}$ *is the output of Algorithm 5.6,*

$$\lim_{n \to \infty} \mathbb{P}(\tau_t \preccurlyeq \tilde{\tau}_t) = 1$$

holds for all $1 \leq t \leq T - 1$.

Proof of Theorem 5.7. The claim is equivalent to

$$\lim_{n \to \infty} \mathbb{P}\left(\tilde{U}_n\right) = 0,$$

where the underestimation event is

$$\tilde{U}_n := \{\tau_t \preccurlyeq \tilde{\tau}_t\}^C.$$

As in the proof of Theorem 4.8, we prove that Lemmas 4.9 and 4.10 still hold in the presence of censoring. For this we first prove that

$$\lim_{n \to \infty} \mathbb{P}\left(\tilde{D}^C_n\right) = 0,$$

where

$$\tilde{D}_n := \left\{\forall w \in \tau^{\mathsf{t}}_t : \tilde{N}_{t+}(w) \geq \tilde{\rho}_n\right\}$$

with

$$\tilde{\rho}_n := \frac{\tilde{b}n}{2}$$

for

$$\tilde{b} := \min_{w \in \tau^{\mathsf{t}}_t} \mathbb{P}\left(X_{t-|w|+1:t} = w,\ J^1_{t-|w|+1:t+1} = (1, ..., 1)\right).$$

Because of (H.3) and (H.4), we can split the condition on X and J^1 into two conditions and obtain

$$\tilde{b} = \min_{w \in \tau_t^{\mathfrak{t}}} \left(\mathbb{P}\left(X_{t-|w|+1:t} = w \right) \cdot j_t(w) \right) > 0,$$

where the positivity is given since $w \in \tau_t^{\mathfrak{t}}$ implies $\mathbb{P}\left(X_{t-|w|+1:t} = w \right) > 0$ and (H.5) gives $j_t(w) > 0$. As in (5.3) we have

$$\mathbb{E}\left[\tilde{N}_{t+}(w) \right] = n j_t(w) \mathbb{P}\left(X_{t-|w|+1:t} = w \right) \geq n\tilde{b}.$$

Note that

$$\left(\tilde{N}_{t+}^i(w) \right)_{1 \leq i \leq n}$$

are independent and identically distributed, since $X^1, ..., X^n$ are and (H.1), (H.2) and (H.4) hold. Now it follows analogously to the proof of Lemma 4.9 that

$$0 \leq \mathbb{P}\left(\tilde{D}_n^C \right) \leq m^t \exp\left(-\frac{\tilde{b}^2}{2} n \right)$$

and thus

$$\lim_{n \to \infty} \mathbb{P}\left(\tilde{D}_n^C \right) = 0.$$

Secondly we have to prove that

$$\lim_{n \to \infty} \mathbb{P}(\tilde{U}_n \cap \tilde{D}_n) = 0$$

holds. By replacing N_t by $\tilde{N}_{t+}$, N_{t+1} by $\tilde{N}_{t+1}$, $\hat{P}(\cdot \mid wu)$ by $\tilde{P}(\cdot \mid wu)$, ρ_n by $\tilde{\rho}_n$, b by $\tilde{b}$, it follows completely analogously to the proof of Lemma 4.10 that

$$0 \leq \mathbb{P}(\tilde{U}_n \cap \tilde{D}_n) \leq 4m^t n^2 e^{-\tilde{b}n\varepsilon}$$

holds. The only argument that needs to be added to the already existing proof is that (H.1)-(H.4) hold. This ensures the independence needed to apply Hoeffding's inequality A.6 in (4.7). Thus

$$\lim_{n \to \infty} \mathbb{P}\left(\tilde{U}_n \right) = 0$$

and the proof is complete, cf. the proof of Theorem 4.8. □

The probability of overestimating the true context tree also still vanishes under censoring.

Theorem 5.8. *Let $X = (X_t)_{1\leq t\leq T} \in \mathcal{P}^{0:T-1}$ with $T \geq 2$ be a tiVLMC with state space M of cardinality $2 \leq m < \infty$ and (unknown) context tree τ. If $X^1_{1:T} = x^1_{1:T}, ..., X^n_{1:T} = x^n_{1:T}$ are n independent observations of X, censored by censoring variables $J^1_{1:T} = j^1_{1:T}, ..., J^n_{1:T} = j^n_{1:T}$ that satisfy (H.1)-(H.5), and $\tilde{\tau}$ is the output of Algorithm 5.6,*

$$\lim_{n\to\infty} \mathbb{P}(\tilde{\tau}_t \preccurlyeq \tau_t) = 1$$

holds for all $1 \leq t \leq T-1$.

Proof of Theorem 5.8. We verify that the arguments used in the proof of Theorem 4.12 are still valid. The simplified overestimation event is

$$\tilde{O}_n = \{\exists w \in \tau_t^{\mathsf{t}}\, \exists u \in M : wu \in \tilde{\tau}_t\}.$$

As in (4.10) we get

$$\mathbb{P}(\tilde{O}_n) \leq \sum_{w\in\tau_t^{\mathsf{t}}} \sum_{u\in M} \mathbb{P}\left(\tilde{\Delta}_t(wu) \geq C\sqrt{n\log n},\, \tilde{N}_{t+}(wu) \geq n_{\min}\right).$$

M is finite and it holds that

$$\tilde{P}_t(x \mid w) = 0 \Rightarrow \tilde{P}_t(x \mid wu) = 0,$$

thus we can apply Lemma 4.13 to obtain

$$\begin{aligned}\Delta_t(wu) &= \mathrm{D}\left[\tilde{P}_t(\cdot \mid wu)||\tilde{P}_t(\cdot \mid w)\right]\tilde{N}_{t+}(wu)\\ &\leq \sum_{x\in M}\left(\frac{\tilde{P}_t(x \mid wu)}{\tilde{P}_t(x \mid w)} - 1\right)\tilde{N}_{t+1}(xwu)\end{aligned}$$

as in (4.11) and (4.12). Note that

$$\tilde{N}_{t+}(wu) = \sum_{x\in M} \tilde{N}_{t+1}(xwu).$$

For arbitrary $z \in M^{1:t-1}$ and arbitrary $x \in M$ the indicators

$$\left(\tilde{N}^i_{t+}(z)\right)_{1\le i\le n} \tag{5.6}$$

are Bin $(1, P_t(z)j_t(z))$-distributed, where $P_t(z) := \mathbb{P}\left(X_{t-|z|+1:t} = z\right)$, cf. (H.1) and (5.4). They are independent, since $X^1, ..., X^n$ are independent and (H.2) and (H.4) hold. The same is true for the Bin $(1, P_{t+1}(xz)j_t(z))$-distributed

$$\left(\tilde{N}^i_{t+1}(xz)\right)_{1\le i\le n}$$

and thus

$$\left(\frac{\tilde{N}^i_{t+}(z) - P_t(z)j_t(z)}{\sqrt{P_t(z)j_t(z)\left(1 - P_t(z)j_t(z)\right)}}\right)_{1\le i\le n}$$

and

$$\left(\frac{\tilde{N}^i_{t+1}(xz) - P_{t+1}(xz)j_t(z)}{\sqrt{P_{t+1}(xz)j_t(z)\left(1 - P_{t+1}(xz)j_t(z)\right)}}\right)_{1\le i\le n}$$

are also independent random variables with zero mean and unit variance. The law of the iterated logarithm A.8 implies

$$\limsup_{n\to\infty} \frac{\tilde{N}_{t+}(z) - nP_t(z)j_t(z)}{\sqrt{P_t(z)j_t(z)\left(1 - P_t(z)j_t(z)\right)}\sqrt{2n\log\log n}} = 1$$

and

$$\limsup_{n\to\infty} \frac{\tilde{N}_{t+1}(xz) - nP_{t+1}(xz)j_t(z)}{\sqrt{P_{t+1}(xz)j_t(z)\left(1 - P_{t+1}(xz)j_t(z)\right)}\sqrt{2n\log\log n}} = 1$$

$\mathbb{P}$-a.s. as in (4.13). Again, let $r := \mathcal{O}\left(\sqrt{n\log\log n}\right)$. Recall the Bachmann-Landau notation, cf. Definition A.9. We write

$$\begin{aligned}
\tilde{N}_{t+1}(xwu) &= nP_{t+1}(xwu)j_t(wu) + r,\\
\tilde{N}_{t+1}(xw) &= nP_{t+1}(xw)j_t(w) + r,\\
\tilde{N}_{t+}(wu) &= nP_t(wu)j_t(wu) + r,\\
\tilde{N}_{t+}(w) &= nP_t(w)j_t(w) + r.
\end{aligned}$$

Now repeat the same computations as in the proof of Theorem 4.12. The quadratic term in (4.14) is now

$$\begin{aligned}&n^2\left(P_{t+1}(xwu)j_t(wu)P_t(w)j_t(w)-P_{t+1}(xw)j_t(w)P_t(wu)j_t(wu)\right)\\=&n^2 j_t(wu)j_t(w)\underbrace{\left(P_{t+1}(xwu)P_t(w)-P_{t+1}(xw)P_t(wu)\right)}_{=0}\end{aligned}$$

and hence still vanishes, even in the presence of censoring. In consequence it immediately follows that

$$\lim_{n\to\infty}\mathbb{P}(\tilde{O}_n)=0$$

by copying the arguments in the proof of Theorem 4.12. This completes the proof. □

We again obtain the correctness of Algorithm 5.6 by combining the Theorems 5.7 and 5.8, for the precise argumentation cf. the proof of Corollary 4.14.

Corollary 5.9. *Let $X=(X_t)_{1\le t\le T}\in\mathcal{P}^{0:T-1}$ with $T\ge 2$ be a tiVLMC with state space M of cardinality $2\le m<\infty$ and (unknown) context tree τ. If $X^1_{1:T}=x^1_{1:T},...,X^n_{1:T}=x^n_{1:T}$ are n independent observations of X, censored by censoring variables $J^1_{1:T}=j^1_{1:T},...,J^n_{1:T}=j^n_{1:T}$ that satisfy (H.1)-(H.5), and $\tilde{\tau}$ is the output of Algorithm 5.6,*

$$\lim_{n\to\infty}\mathbb{P}(\tau_t=\tilde{\tau}_t)=1$$

holds for all $1\le t\le T-1$.

5.3 Pruning with an arbitrary norm

For

$$\tilde{\Delta}_t^{L^1}(wu):=\tilde{N}_{t+}(wu)\|\tilde{P}_t(\cdot\mid wu)-\tilde{P}_t(\cdot\mid w)\|_1^2$$

the censored version

$$2\tilde{\Delta}_t(wu)\ge\tilde{\Delta}_t^{L^1}(wu)$$

of (4.15) still holds by Proposition 4.3. Thus it immediately follows that Algorithm 5.6 remains correct if one replaces $\tilde{\Delta}_t$ by $\tilde{\Delta}_t^{L^1}$, i.e. if one prunes with the L^1-norm instead of the Kullback-Leibler divergence, cf. the proofs of Theorem 4.16 and Theorem 4.17 and the resulting Corollary 4.18. Following the arguments

of Chapter 4.3, it is trivial that we can again use any norm and thus the following corollary holds.

Corollary 5.10. *Let $X = (X_t)_{1\leq t\leq T} \in \mathcal{P}^{0:T-1}$ with $T \geq 2$ be a tiVLMC with state space M of cardinality $2 \leq m < \infty$ and (unknown) context tree τ. If $X^1_{1:T} = x^1_{1:T}, ..., X^n_{1:T} = x^n_{1:T}$ are n independent observations of X, censored by censoring variables $J^1_{1:T} = j^1_{1:T}, ..., J^n_{1:T} = j^n_{1:T}$ that satisfy (H.1)-(H.5), and $\tilde{\tau}$ is the output of Algorithm 5.6, where $\tilde{\Delta}_t(wu)$ is replaced by*

$$\tilde{\Delta}_t^{\|\cdot\|}(wu) := \tilde{N}_{t+}(wu) \left\| \tilde{P}_t(\cdot \mid wu) - \tilde{P}_t(\cdot \mid w) \right\|^2$$

and $\|\cdot\|$ is an arbitrary norm defined on the space of all functions $f : M \to \mathbb{R}$,

$$\lim_{n\to\infty} \mathbb{P}(\tilde{\tau}_t = \tau_t) = 1$$

holds for all $1 \leq t \leq T-1$.

Corollary 5.10 says that Algorithm 5.6 outputs a consistent estimate of the true context tree, regardless of which norm is used for the censored time-sensitive pruning measure. This is the analogue to Theorem 4.19.

5.4 Model limitations

While (H.1), (H.2) and (H.5) are natural assumptions, (H.3) and (H.4) are not always justified. As a matter of fact, insurance holders often decide to surrender their contract (and exit the observation) because of their past experience or they decide to surrender their contract together, e.g. if they are in a relationship. An invalid insurance holder who receives a disability pension will most likely not cancel his or her long-term care insurance. On the contrary, a person with a perfect health history might feel that the long-term care insurance is not economically beneficial and thus terminates the contract.
Lapsing contracts due to one's own experience can be modelled by allowing the censoring variable J^i to depend on the observation X^i, $i = 1, ..., n$, i.e. by not assuming (H.3). This comes at a price: $w \in \tau_t^{\mathsf{t}}$ implies that

$$\mathbb{P}(X_{t-|w|+1:t} = w) > 0$$

holds and (H.5) says that

$$j_t(w) = \mathbb{P}\left(J^i_{t-|w|+1:t+1} = (1, ..., 1)\right) > 0$$

holds. Together with the independence (H.3) of J^i and X^i, (H.4) of J^j and X^i for $i \neq j$ and the fact that $X^1, ..., X^n$ are distributed as X this implies that

$$\mathbb{P}\left(X_{t-|w|+1:t} = w,\ J^i_{t-|w|+1:t+1} = (1, ..., 1)\right) > 0, \tag{5.7}$$

i.e. that there is a positive possibility to observe everything relevant. Thus under (H.3) and (H.4), (H.5) is sufficient to guarantee (5.7). If we do not assume (H.3) and (H.4), we have to directly assume the stronger version (5.7) of (H.5):

(H.6) For every $1 \leq t \leq T-1$ and every $w \in \text{tpsupp}_t(X)$

$$\mathbb{P}\left(X_{t-|w|+1:t} = w,\ J^i_{t-|w|+1:t+1} = (1, ..., 1)\right) > 0,$$

$i = 1, ..., n$, holds.

However, in the following we will see that in general the results of the Chapters 5.1-5.3 cannot be obtained if one only assumes (H.1), (H.2), (H.4) and (H.6) instead of (H.1)-(H.5). Hence the proposed model heavily relies on independence between the censoring variable and its observation, i.e. on assumption (H.3). This problem is illustrated by an explicit example in the later Chapter 8.1.4.
First we disprove Proposition 5.3, i.e. we prove that the estimator for the transition probabilities is not necessarily unbiased any more. Note that, while assumption (H.3) is sufficient to prove that the estimator is unbiased, cf. Proposition 5.3, (H.3) being violated is not sufficient to prove that the estimator is biased. We answer the question whether X^i is censored or uncensored in a time frame of interest by aggregating information carried by J^i: Are all coordinates in the time frame one (uncensored) or are some of them zero (censored)? Even if X^i and J^i are dependent, it is possible (but should be expected to be rare in real world applications) that the dependency is lost in the aggregation process. Hence in the following Proposition 5.11 we assume that the dependency survives the aggregation in order to prove that, then inevitably, our estimated transition probabilities are biased. The message of Proposition 5.11 to the reader is that if (H.3) is violated, one has to expect a biased estimation.

Proposition 5.11. *Assume (H.1) and (H.4). Assume that (H.3) is violated in the sense that there exist a* $1 \leq t \leq T-1$, $x \in M$ *and* $w \in M^{1:t}$ *with* $\tilde{N}_{t+}(w) > 0$ *such that*

$$\mathbb{P}\left(X^i_{t-|w|+1:t+1} = xw \,\middle|\, X^i_{t-|w|+1:t} = w,\ \mathbb{1}\left(J^i_{t-|w|+1:t+1} = (1,...,1)\right) = 1\right) \neq P_t(x \mid w).$$

Then

$$\mathbb{E}\left[\tilde{P}_t(x \mid w)\right] \neq P_t(x \mid w),$$

i.e. $\tilde{P}_t(x \mid w)$ *is biased.*

Proof of Proposition 5.11. If

$$\tilde{N}^i_{t+}(w) = N^i_t(w) \cdot \mathbb{1}\left(J^i_{t-|w|+1:t+1} = (1,...,1)\right) = 1$$

holds, it is implied that

$$\mathbb{1}\left(J^i_{t-|w|+1:t+1} = (1,...,1)\right) = 1,$$

since both factors are binary. Thus

$$\begin{aligned}
&\mathbb{E}\left[\tilde{N}^i_{t+1}(xw) \,\middle|\, \tilde{N}^i_{t+}(w) = 1\right] \\
=&\mathbb{E}\left[N^i_{t+1}(xw) \cdot \mathbb{1}\left(J^i_{t-|w|+1:t+1} = (1,...,1)\right) \,\middle|\, \tilde{N}^i_{t+}(w) = 1\right] \\
=&\mathbb{E}\left[N^i_{t+1}(xw) \cdot 1 \,\middle|\, \tilde{N}^i_{t+}(w) = 1\right] \\
=&\mathbb{P}\left(X^i_{t-|w|+1:t+1} = xw \,\middle|\, X^i_{t-|w|+1:t} = w,\ \mathbb{1}\left(J^i_{t-|w|+1:t+1} = (1,...,1)\right) = 1\right) \\
\neq&P_t(x \mid w).
\end{aligned}$$

Repeating the calculations within the proof of Theorem 2.12, we thus get that

$$\begin{aligned}
\mathbb{E}\left[\tilde{P}_t(x \mid w)\right] &= \mathbb{P}\left(X^i_{t-|w|+1:t+1} = xw \,\middle|\, X^i_{t-|w|+1:t} = w,\ J^i_{t-|w|+1:t+1} = (1,...,1)\right) \\
&\neq P_t(x \mid w)
\end{aligned}$$

holds. □

In general the estimator does also not preserve the consistency stated in Proposition 5.4 if (H.3) is false, as will be shown next.

Proposition 5.12. *Assume (H.1), (H.2), (H.4) and (H.6). Assume that (H.3) is violated in the sense that there exist a* $1 \leq t \leq T-1$, $x \in M$ *and* $w \in M^{1:t}$ *with* $\tilde{N}_{t+}(w) > 0$ *such that*

$$\mathbb{P}\left(X^i_{t-|w|+1:t+1} = xw \,\middle|\, X^i_{t-|w|+1:t} = w,\ \mathbb{1}\left(J^i_{t-|w|+1:t+1} = (1,...,1)\right) = 1\right) \neq P_t(x \mid w).$$

Then

$$\max\left|\tilde{P}_t(x \mid w) - P_t(x \mid w)\right| \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} c > 0,$$

where the maximum runs over all well defined transition probabilities, i.e. over $1 \leq t \leq T-1$, $x \in M$ *and* $w \in M^{1:t}$ *with* $\mathbb{P}(X_{t-|w|+1:t} = w) > 0$. *Thus in particular*

$$\max\left|\tilde{P}_t(x \mid w) - P_t(x \mid w)\right| \overset{n\to\infty}{\underset{\mathbb{P}-\text{a.s.}}{\not\to}} 0$$

holds.

Proof of Proposition 5.12. As in (5.6) the indicators

$$\left(\tilde{N}^i_{t+1}(xw)\right)_{1\leq i\leq n}$$

are $\text{Bin}\left(1, \mathbb{P}\left(X_{t-|w|+1:t+1} = xw,\ J^i_{t-|w|+1:t+1} = (1,...,1)\right)\right)$-distributed and independent. Thus the strong law of large numbers A.1 again implies

$$\frac{\tilde{N}_{t+1}(xw)}{n} \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} \mathbb{P}\left(X_{t-|w|+1:t+1} = xw,\ J^i_{t-|w|+1:t+1} = (1,...,1)\right) =: a$$

and analogously

$$\frac{\tilde{N}_{t+}(w)}{n} \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} \mathbb{P}\left(X_{t-|w|+1:t} = w,\ J^i_{t-|w|+1:t+1} = (1,...,1)\right) =: b.$$

Assumption (H.6) guarantees that both limits, a and b, are positive, therefore

$$\tilde{P}_t(x \mid w) = \frac{\tilde{N}_{t+1}(xw)}{n}\frac{n}{\tilde{N}_{t+}(w)} \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} \frac{a}{b} > 0.$$

Since $\mathbb{P}$-a.s. limits are unique, the proof is complete if

$$\frac{a}{b} \neq P_t(x \mid w).$$

By Bayes' theorem A.12, for

$$H := \Big\{X_{t-|w|+1:t} = w,\ J^i_{t-|w|+1:t+1} = (1,...,1)\Big\},$$

we get

$$\begin{aligned}\frac{a}{b} =& \frac{\mathbb{P}\left(X_{t-|w|+1:t+1} = xw,\ J^i_{t-|w|+1:t+1} = (1,...,1)\right)}{b}\\ =& \frac{\mathbb{P}\left(X_{t-|w|+1:t+1} = xw,\ J^i_{t-|w|+1:t+1} = (1,...,1) \,\middle|\, H\right)}{b} \cdot b\\ &+ \frac{\mathbb{P}\left(X_{t-|w|+1:t+1} = xw,\ J^i_{t-|w|+1:t+1} = (1,...,1) \,\middle|\, H^C\right)}{b} \cdot (1-b).\end{aligned}$$

Now note that the second fraction vanishes completely, since

$$\Big\{X_{t-|w|+1:t+1} = xw\Big\} \cap \Big\{X_{t-|w|+1:t} \neq w\Big\} = \emptyset$$

and

$$\Big\{J^i_{t-|w|+1:t+1} = (1,...,1)\Big\} \cap \Big\{J^i_{t-|w|+1:t+1} \neq (1,...,1)\Big\} = \emptyset$$

and thus its nominator is zero by De Morgan's laws A.13. Hence

$$\begin{aligned}\frac{a}{b} &= \mathbb{P}\left(X_{t-|w|+1:t+1} = xw,\ J^i_{t-|w|+1:t+1} = (1,...,1) \,\middle|\, H\right)\\ &= \frac{\mathbb{P}\left(\Big\{X_{t-|w|+1:t+1} = xw\Big\} \cap H\right)}{\mathbb{P}(H)}\\ &= \frac{\mathbb{P}\left(X_{t-|w|+1:t+1} = xw\,, X_{t-|w|+1:t} = w,\ J^i_{t-|w|+1:t+1} = (1,...,1)\right)}{\mathbb{P}\left(X_{t-|w|+1:t} = w,\ J^i_{t-|w|+1:t+1} = (1,...,1)\right)}\\ &= \mathbb{P}\left(X_{t-|w|+1:t+1} = xw \,\middle|\, X_{t-|w|+1:t} = w,\ J^i_{t-|w|+1:t+1} = (1,...,1)\right)\\ &= \mathbb{P}\left(X_{t-|w|+1:t+1} = xw \,\middle|\, X_{t-|w|+1:t} = w,\ \mathbb{1}\left(J^i_{t-|w|+1:t+1} = (1,...,1)\right) = 1\right)\\ &\neq P_t(x \mid w).\end{aligned}$$

□

As seen in the proof of Proposition 5.12, we asymptotically base our decision on whether to keep the leaf wu or prune it down to w on the distance measured between

$$\mathbb{P}\left(X^i_{t+1} = x \,\middle|\, X^i_{t-|wu|+1:t} = wu,\ J^i_{t-|wu|+1:t+1} = (1,...,1)\right)$$

and

$$\mathbb{P}\left(X^i_{t+1} = x \,\middle|\, X^i_{t-|w|+1:t} = w,\ J^i_{t-|w|+1:t+1} = (1, ..., 1)\right).$$

If assumption (H.3) is true the conditioning on the censoring variable J^i becomes redundant and both probabilities become transition probabilities: Now we asymptotically compare

$$P_t\left(x \mid wu\right) = \mathbb{P}\left(X^i_{t+1} = x \,\middle|\, X^i_{t-|wu|+1:t} = wu\right)$$

and

$$P_t\left(x \mid w\right) = \mathbb{P}\left(X^i_{t+1} = x \,\middle|\, X^i_{t-|w|+1:t} = w\right)$$

as it was our initial intention. Here we know that we infer the true context tree, cf. Corollary 5.9. In the Propositions 5.11 and 5.12 we have learned that this is not necessarily the case if (H.3) is violated.

If we calculate our estimate on the base of the second comparison, we infer a context tree that displays the behaviour of our collective of insured explained by the information carried by X^i up to time t. If e.g. X^i documents the health status of the insured, we infer their behaviour due to changes in their previous health development. However, when we have to base our decision on the first comparison, we lose the ability to credit their behaviour to their health: Now we cannot differentiate whether the reason for the behaviour of the collective is encoded in X^i or J^i up to time t or (most likely) a mixture of both information sources.

The advantage here is that in insurance practice censoring mostly displays "not being insured", e.g. when a contract has ended, has not yet started or is cancelled. If our main goal is to calculate net premiums or prospective reserves, cf. Chapter 9, we need to analyse how our collective of insured behaves: It is this behaviour that triggers payments between the insurance company and the insurance holders and thus determines e.g. the net premium. To perform the calculations, the knowledge about how the collective behaves is essential, the reasons why it behaves in that particular way are not essential. I.e. mathematically it is not of great harm that we cannot differentiate whether the reason for the behaviour is encoded in X^i, J^i or a mixture of both up to time t. An estimated context tree calculated under a violated assumption (H.3) still contains sufficient information.

The described censoring procedure can also be interpreted as the introduction of a new state c that indicates being censored. When estimating transition probabilities we then only consider transitions between the original states in M and not in $M \cup \{c\}$.

6 Tuning the algorithm

Algorithm 4.7 and its censored generalization Algorithm 5.6 both require one tuning parameter as an input: the cutoff value C. The consistency result Corollary 5.9 states that for every possible value $C > 0$ of the cutoff parameter, the estimated context tree $\tilde{\tau}$ in probability converges to the true context tree τ of the tiVLMC as the number of observations n increases. Since in practice one is usually not able to simply increase the amount of available data, it remains an open question which choice of C yields the best fitting model for a fixed n.
Gladly, since C is a one-dimensional, positive parameter, tuning the algorithm for C is a straightforward procedure: Within the classical VLMC setting presented in the introductory Chapter 1.2 the authors of Mächler and Bühlmann (2004, page 443) state that "[...] optimization with respect to the cutoff is relatively easy" while "optimizing among all subtrees from a large tree is prohibitive". This statement does not only remain true in the tiVLMC-setting, it is of even higher significance, since one would not only have to optimize among all subtrees from one large tree, but from $T-1$ large trees due to the time-inhomogeneity.
The larger C, the smaller are the context tree estimates output by the algorithm. I.e.

$$\tilde{\tau}_t \preccurlyeq \tilde{\tau}'_t \tag{6.1}$$

holds for all $t = 1, ..., T-1$ if $C > C'$ and $\tilde{\tau}$ was obtained by applying Algorithm 5.6 with the cutoff C and $\tilde{\tau}'$ with the cutoff C'. Thus running the algorithm multiple times with increasing cutoff values leads to a candidate set of decreasing models. Then one can solve the model selection problem with the use of information criteria, cross-validation (cf. Larson (1931) and Jones (1987) for the early origins of cross-validation) or other model selection techniques known in statistics.
Within this chapter we are going to take a close look at the two most commonly used information criteria: Akaike's information criteria (abbreviated by "AIC"), cf. Akaike (1998), and the Bayesian information criteria (abbreviated by "BIC"), cf. Schwarz et al. (1978). And we will see what they explicitly look like for tiVLMC. Choosing the model with the lowest AIC-value from a set of candidate models is well-known to minimize the Kullback-Leibler divergence D, cf. Definition 4.1, between the selected and the true model, e.g. cf. Claeskens et al. (2008). Therefore the AIC is a natural choice to tune Algorithm 5.6: The pruning measure in the original Algorithm 1.11 from Rissanen (1983) is based on

the Kullback-Leibler divergence. Both, AIC and BIC, consist of two main components: the log-likelihood function and the model complexity.

6.1 The log-likelihood function

As we have already seen in (4.1), the estimated log-likelihood function of observing $x^1_{1:T}, ..., x^n_{1:T}$ (conditioned on the first state) based on the estimated context function $\tilde{c}$ is

$$\begin{aligned} \log L(\tilde{c}) &= \log\left(\prod_{i=1}^{n}\prod_{t=1}^{T-1} \tilde{P}_t\left(x^i_{t+1} \,\middle|\, \tilde{c}_t(x^i_{1:t})\right)\right) \\ &= \sum_{i=1}^{n}\sum_{t=1}^{T-1} \log \tilde{P}_t\left(x^i_{t+1} \,\middle|\, \tilde{c}_t(x^i_{1:t})\right). \end{aligned} \tag{6.2}$$

Since we can always compute $\tilde{\tau}$ from $\tilde{c}$ and vice versa, one can also talk about the estimated log-likelihood function based on the estimated context tree,

$$\log L(\tilde{\tau}) := \log L(\tilde{c}).$$

In certain scenarios this notation is more convenient. We can even perceive the estimated log-likelihood function as a function of the cutoff C by setting

$$\log L(C) := \log L(\tilde{c})$$

with $\tilde{c}$ being the estimated context function obtained by running Algorithm 5.6 with the cutoff C.

6.2 The model complexity

As stated in Proposition 3.9, every tiVLMC $X \in \mathcal{P}^{0:T-1}$ with context function c is uniquely determined by the marginal distribution $\mathbb{P}^{X_1}$ and its set of transition probabilities

$$\mathbb{P}_c = \{P_t(x \,|\, c_t(w)) : x \in M, 1 \leq t \leq T-1, w \in \mathrm{tpsupp}_t(X)\},$$

cf. Definition 3.8. The initial distribution $\mathbb{P}^{X_1}$ does not depend on the context function c and hence is totally irrelevant for model selection, i.e. the question how to choose c. Thus the model complexity of a tiVLMC, i.e. the number of free

model parameters, equals the minimal number of parameters sufficient to fully describe $\mathbb{P}_c$. It holds that

$$\begin{aligned}|\mathbb{P}_c| &= \sum_{t=1}^{T-1} \sum_{x \in M} |\{P_t(x \mid c_t(w)) : w \in \text{tpsupp}_t(X)\}| \\ &= m \sum_{t=1}^{T-1} |\{P_t(x \mid c_t(w)) : w \in \text{tpsupp}_t(X)\}| \\ &= m \sum_{t=1}^{T-1} |\text{image}(c_t)| \\ &= m \sum_{t=1}^{T-1} |\tau_t|.\end{aligned}$$

If we fit a tiVLMC, we do not need to estimate the transition probabilities $P_t(x \mid c_t(w))$ for all $x \in M$. It is sufficient to estimate $P_t(x \mid c_t(w))$ for all but one $x_0 \in M$. These estimates then determine

$$P_t(x_0 \mid c_t(w)) = 1 - \sum_{\substack{x \in M \\ x \neq x_0}} P_t(x \mid c_t(w)).$$

Thus the minimal number of parameters sufficient to uniquely describe $\mathbb{P}_c$, i.e. the model complexity of the tiVLMC, amounts to

$$(m-1) \sum_{t=1}^{T-1} |\tau_t|.$$

This is in line with Mächler and Bühlmann (2004, page 442-443).

6.3 Two information criteria

Varying the tuning parameter C leads to different estimates of the context tree $\tilde{\tau}$ (and thus to different estimates $\tilde{c}$ of the true context function c). Thus AIC and BIC, viewed as functions of the cutoff C, take the form

$$\begin{aligned}\text{AIC}(C) &= -2 \log L(\tilde{\tau}) + 2(m-1) \sum_{t=1}^{T-1} |\tilde{\tau}_t|, \\ \text{BIC}(C) &= -2 \log L(\tilde{\tau}) + \log n \cdot (m-1) \sum_{t=1}^{T-1} |\tilde{\tau}_t|.\end{aligned} \tag{6.3}$$

Then we tune the model fitting for C by choosing

$$\begin{aligned} C_{\text{AIC}} &= \arg\min_{C>0} \text{AIC}(C), \\ C_{\text{BIC}} &= \arg\min_{C>0} \text{BIC}(C) \end{aligned} \tag{6.4}$$

as the cutoff value C depending on whether AIC or BIC is preferred. Since $\mathbb{R}_{>0}$ is a set of infinite cardinality (it even is uncountable), it is numerically impossible to calculate $\text{AIC}(C)$ or $\text{BIC}(C)$ for all $C > 0$ in finite time. Therefore the user who performs the tuning has to specify a finite subset $\mathscr{C} \subset \mathbb{R}_{>0}$ and use

$$C_{\text{AIC}} = \arg\min_{C\in\mathscr{C}} \text{AIC}(C)$$

or

$$C_{\text{BIC}} = \arg\min_{C\in\mathscr{C}} \text{BIC}(C)$$

instead of (6.4). Typically it makes sense to choose $\mathscr{C}$ as a lattice of the interval $[\varepsilon, E]$ with ε close to zero and E big enough so that the model fitted using E as the cutoff is the total independence model (i.e. $\tilde{\tau}_t = \{e\}$ for every $t = 1, ..., T-1$). Tuning with cutoffs $C > E$ comes with no benefit.

Two artificial examples of the tuning procedure are given in Chapter 8.2, the first subchapter working in an uncensored data environment, whereas censoring is present in the second subchapter.

7 Implementation of the algorithm

We implement Algorithm 5.6 and the tuning procedure described in Chapter 6 in R (version 3.5.2), cf. R Core Team (2018).
We use the data.tree-package, which offers some general tools for working with hierarchical data, cf. Glur (2018). We make use of the knitr-package to visually display R code, cf. Xie (2013). The tictoc-package offers functions for time measurements, cf. Izrailev (2014). To speed up computations, the doMPI- and foreach-package, cf. Weston (2013) respectively Analytics and Weston (2015), are used to parallelize code.

7.1 The estimateTau-function

```
estimateTau <- function(data, ties=NULL, from=NULL, to=NULL,
                        nmin=2, max.length=NULL, cutoff, measure,
                        log.like=F, tau.max=NULL,
                        calculate.tau=T) {
  #extracts the state space from the data, NA aka censoring is
  #not counted
  alphabet <- na.omit(unique(c(data)))
  #throw an error if one state is labeled "e"
  if ("e" %in% alphabet) {
    stop("e is not an allowed state name as it denotes the empty
         string! Rename e within the data.")
  }
  #ties is an optional argument with default value NULL
  #throw an error if the frequencies of the observations do not
  #match the number of the observations
  if ((length(ties) != dim(data)[1]) & (!is.null(ties))) {
    stop("length(ties) and dim(data)[1] have to be equal!")
  }
  #from is an optional argument with default value 1
  if (is.null(from)) from <- 1
  #to is an optional argument with the latest time possible as
  #default
  if (is.null(to)) to <- dim(data)[2]-1
```

```
#the following boolean saves whether tau.max was input
tau.max.input  <- ifelse(is.null(tau.max), F, T)
#if tau.max was input, it has to be of correct length
if (tau.max.input & (length(from:to) != length(tau.max))) {
  stop("tau.max has incorrect length.")
}
#if tau.max was not input, we initialize it
if (!tau.max.input) tau.max <- list()
#in this list the tau at t will be saved
tau <- list()
#number of observations
nobs <- dim(data)[1]
#calculating tau.max and tau for all t
for (time in from:to) {
  #we measure the time needed for each iteration and
  #update the user about the progress
  msg <- paste(time-from+1, "out of", to-from+1,
               "context trees estimated")
  tic(msg=msg)
  #calculate tau.max for the current t if it was not input
  if (!tau.max.input) {
    tau.max[[time]] <- buildTauMax(data=data, ties=ties,
                                   time=time, nmin=nmin,
                                   max.length=max.length,
                                   alphabet=alphabet)
  }
  #if calculate.tau=T, we calculate tau by pruning tau.max
  #down to tau at the current t, if tau=F, only tau.max is
  #output
  if (isTRUE(calculate.tau)) {
    tau[[time]] <- pruneTauMax(tau.max[[time]], cutoff=cutoff,
                               measure=measure,
                               alphabet=alphabet,
                               nobs=nobs)
  }
  toc()
}
```

```
  if (isTRUE(calculate.tau)) {
    #if log.like is true, we perform the calculation of
    #the loglikelihood function
    log.like <- ifelse(isTRUE(log.like),
                       calculateLogLikelihood(data=data, ties=ties,
                                              tau=tau), NA)
    #we calculate the model complexity of the fitted tiVLMC
    model.complexity <- calculateModelComplexity(alphabet=alphabet,
                                                 tau=tau)
  } else {
    log.like <- NA
    model.complexity <- NA
  }
  #we save the estimated tau, the tau.max, the call, the
  #model complexity and the value of the log-likelihood function
  estimate <- list(call=match.call(), tau=tau, tau.max=tau.max,
                   model.complexity=model.complexity,
                   log.like=log.like)
  return(estimate)
}
```

R-Code 7.1: The estimateTau-function

The estimateTau-function shown in R-Code 7.1 wraps up the whole algorithm. It takes the training data data as its first input. data has to be of class matrix with entries of type character. Censoring can be represented by filling the censored values with NAs.

The optional ties argument lets the user input an integer vector of the frequencies of the observations. If there are ties in the data, i.e. the same observation is made multiple times, it is more efficient to list that observation only once in the data and handover its frequency by inputting ties. The i-th element of ties denotes the frequency of the i-th observation (i.e. the i-th row in data). The length of ties needs to equal the number of observations, i.e. its length needs to equal $n =$ dim(data)[1]. If ties is not input, it is defaulted to NULL and all calculations will assume that each observation, i.e. each row in data, was observed once.

The third and fourth input, from and to, let the user specify the time points of interest, i.e. the range of t for which $\tilde{\tau}_t$ shall be computed. Both values have to be

integers and naturally from $\leq$ to has to hold. The timespan of maximal possible length is taken as the default.
Via nmin the user can specify the lower bound $n_{\min}$. If the number of occurrences $\tilde{N}_{t+}(w)$ of a sequence w falls below $n_{\min}$, it is not included in $\tau^{\mathsf{t}}_{\max}$ in the first step of Algorithm 5.6. The default value of nmin is two. This is in accordance with Bühlmann et al. (1999, page 490), who state in their Remark 3.5 that "[t]he number two is a low enough value in practice which guarantees a sufficiently large initial tree and at least two observations to estimate transition probabilities associated with the terminal nodes [...]" and also with Weinberger et al. (1995, page 646).
By setting the max.length input, one can limit the depth of the trees $\tilde{\tau}_t$, i.e. limit the modelled length of dependency into the past. max.length needs to be an integer with max.length ≥ 1. If max.length is not chosen, the default value used is the full range possible. Limiting the maximal context length reduces the needed computation time by a large extent. The downside is that, if

$$\text{max.length} < \max_{1 \leq t \leq T-1} |c_t|$$

is chosen, i.e. the maximal allowed length of estimated contexts is smaller than the order of the data generating tiVLMC, the output of the estimateTau-function cannot be consistent any more, Corollary 5.9 is violated. If

$$\text{max.length} \geq \max_{1 \leq t \leq T-1} |c_t|,$$

there is no conflict with Corollary 5.9.
cutoff lets one choose the cutoff constant $C > 0$ and via measure one can choose whether the Kullback-Leibler divergence ("KLD") or the L^1-norm ("L1") is used for pruning $\tau^{\mathsf{t}}_{\max}$ down to $\tilde{\tau}_t$.
If the user inputs log.like as TRUE, the value of the estimated log-likelihood function of the fit is computed and stored in the output. This input is optional, not specifying it or defining it as FALSE decreases the needed computation time.
As explained in Chapter 6, one often fits tiVLMC-models for different values of the cutoff parameter in order to tune the model. Step one of the performed Algorithm 5.6 is not depending on the cutoff, i.e. $\tau^{\mathsf{t}}_{\max}$ is not depending on the cutoff. Thus it is efficient to only compute $\tau^{\mathsf{t}}_{\max}$ once and repeat only step two of Algorithm 5.6 multiple times, i.e. the pruning procedure. To enable such behaviour, estimateTau lets the user input (the already computed) $\tau^{\mathsf{t}}_{\max}$ via the tau.max-argument. If tau.max is not specified, it defaults to NULL and both algorithmic

steps one and two are performed as usual.
The estimateTau-function outputs the model fit, i.e. the estimated context trees in the list tau. Additionally the unpruned trees are stored in the list tau.max and the number of model parameters model.complexity is also saved. If log.like was set to TRUE, it stores the value of the estimated log-likelihood function, otherwise it is NA.

7.2 The buildTauMax-function

```
buildTauMax <- function(data, ties, time, nmin, max.length,
                        alphabet) {
  #defining the default-value of max.length
  if (is.null(max.length)) max.length <- time
  #time period relevant for the following calculations
  earlier.time <- max(time-max.length+1, 1)
  later.time <- time+1
  #the data that is relevant to built tau.max
  relevant.data <- data[, earlier.time:later.time]
  #initialize the root node
  tau.max <- initializeRootNode(futures=data[,later.time],
                                ties=ties, alphabet=alphabet)
  #we run through every observation and separate it...
  for (i in 1:dim(relevant.data)[1]) {
    #... into the past of maximal length...
    past <- relevant.data[i, 1:(dim(relevant.data)[2]-1)]
    #... and the (t+1)-th state, i.e. the future
    future <-  relevant.data[i, dim(relevant.data)[2]]
    #we save the frequency of the observation if the ties argument
    #was input, otherwise the frequency is one
    freq <- ifelse(!is.null(ties), ties[i], 1)
    #we add the branch to the tree
    #here we take the frequency of the observation into regard
    tau.max <- addBranch(tree=tau.max, past=past, future=future,
                         freq=freq, alphabet=alphabet)
  }
  #prune every node with an occurrence counter smaller than nmin
```

```
  Prune(tau.max, function(node) node$counter>=nmin)
  return(tau.max)
}
```

R-Code 7.2: The buildTauMax-function

The buildTauMax-function shown in R-Code 7.2 carries out step one of Algorithm 5.6 for a fixed time $= t$.
The root node is initialized by calling the initializeRootNode-function. Then we iterate through every observation and add the sequence to the tree tau.max by calling the addBranch-function. For every sequence w, addBranch also keeps track of its frequency $\tilde{N}_{t+}(w)$. Doing this, we get a tree that contains all sequences of specified length. Finally, we iteratively prune the leaves of tau.max, so that all remaining sequences w fulfil the inequality $\tilde{N}_{t+}(w) \geq$ nmin.
Then the pruned tree is exactly $\tau_{\max}$.

7.2.1 The initializeRootNode-function

```
initializeRootNode <- function(futures, ties, alphabet) {
  #we initialize the tree and call the root node "e"
  root.node <- Node$new("e")
  #we initialize the counter of the empty string "e" to be all
  #observations but those censored at t+1
  #the frequencies of the observations are taken into account
  root.node$counter <- ifelse(is.null(ties),
                              length(na.omit(futures)),
                              sum(ties[which(!is.na(futures))]))
  #initialize and name the transition counter
  transitions <- rep(0, times=length(alphabet))
  names(transitions) <- alphabet
  #count how often each state is a future
  for (state in alphabet) {
    transitions[state] <- ifelse(is.null(ties),
                                 sum(na.omit(futures)==state),
                                 sum(ties[which(na.omit(futures)
                                                ==state)]))
```

```
  }
  #save the transition counts
  root.node$transitions <- transitions
  return(root.node)
}
```

R-Code 7.3: The initializeRootNode-function

The initializeRootNode-function, cf. R-Code 7.3, creates the root node of the tree, which is always the empty string e.
We assign two attributes to every node in a tree (i.e. to every sequence w): The counter-attribute which saves the value $\tilde{N}_{t+}(w)$ and the transitions-attribute, which saves the values $\tilde{N}_{t+1}(xw)$ for all $x \in M$. Thus every node w carries sufficient information to calculate the transition probabilities $\tilde{P}_t(x \mid w)$.
For the root node e we have that counter $= \tilde{N}_{t+}(e)$ equals the number of uncensored observations at $t+1$ and transitions["x"] $= \tilde{N}_{t+1}(xe)$ equals the number of observations which are in x at time $t+1$. This information is retrieved from the input futures, which contains the observations $X_{t+1}^1, ..., X_{t+1}^n$ and ties, which contains the frequencies of the observations.

7.2.2 The addBranch-function

```
addBranch <- function(tree, past, future, freq, alphabet) {
  #if the future is censored, the observation does not contribute
  #and we return the input tree
  if(is.na(future)) return(tree)
  #we start running down the tree at the root
  node <- tree
  #this loop navigates us down the tree
  for (i in length(past):1) {
    state <- past[i]
    #if the first NA is encountered, longer pasts are not
    #considered and we exit
    if(is.na(state)) return(tree)
    #we move one step deeper in the tree
    child <- node[[state]]
```

```
    #if the child is not already existing...
    if(is.null(child)) {
      #... we initialize it with the correct counts
      node <- initializeNode(node=node, state=state,
                             future=future, freq=freq,
                             alphabet=alphabet)
    #if the child is already existing...
    } else {
      #... we update its counts
      node <- updateNode(node=node, state=state,
                         future=future, freq=freq)
      #both, initializeNode and updateNode, update node <- child
      #for the next iteration of the loop
    }
  }
  return(tree)
}
```

R-Code 7.4: The addBranch-function

Via calling the addBranch-function, shown in R-Code 7.4, we add branches, i.e. sequences, to the tree.
Within every iteration of the for-loop in the buildTauMax-function, cf. R-Code 7.2, we take a look at one sequence w that needs to be added to the tree. The addBranch-function first checks whether the current observation is censored at $t+1$. If so, this observation is not considered because it does not contribute to $\tilde{N}_{t+}(w)$ nor to $\tilde{N}_{t+1}(w)$. We run through the sequence, i.e. run down the tree, and increase the counter-attribute of every visited node by the frequency of this sequence and update the transitions-attribute by calling the updateNode-function. If the complete sequence is not already existing in the tree, the missing nodes are added and their attributes are initialized by calling initializeNode.

7.2.3 The initializeNode-function

```
initializeNode <- function(node, state, future, freq, alphabet) {
  #initialize the missing node/the new child
  node$AddChild(state)
  #switch over to the new child
  node <- node[[state]]
  #initialize its counter to be freq
  node$counter <- freq
  #initialize the transition counter
  transitions <- rep(0, times=length(alphabet))
  names(transitions) <- alphabet
  #initialize the transition counter of the current future to freq
  transitions[future] <- freq
  node$transitions <- transitions
  return(node)
}
```

R-Code 7.5: The initializeNode-function

7.2.4 The updateNode-function

```
updateNode <- function(node, state, future, freq) {
  #switch over to the child
  node <- node[[state]]
  #increase its counter by freq
  node$counter <- node$counter+freq
  #increase its transition counter of the current future by freq
  node$transitions[future] <- node$transitions[future]+freq
  return(node)
}
```

R-Code 7.6: The updateNode-function

7.3 The pruneTauMax-function

The buildTauMax-function explained in the previous Chapter 7.2 outputs $\tau_{\max}$ and hence executes the first step of Algorithm 5.6. What is left to do, is pruning the tree $\tau_{\max}$ down to $\tilde{\tau}_t$, i.e. performing the second step of Algorithm 5.6. The data.tree-package, again cf. Glur (2018), offers the handy function Prune for pruning R-objects of type Node. We already used Prune in R-Code 7.2.

```
pruneTauMax <- function(tau.max, cutoff, measure, alphabet, nobs) {
  #we calculate the pruning measure divided by sqrt(nlogn) for
  #every node of tau.max, i.e. we calculate the maximal cutoff
  #value so that the node is not pruned
  tau.max$Do(function(node) {node$criticalCutoff <-
    calculatePruningMeasure(node, measure)})
  #we now clone tau.max, because we want to save tau.max and
  #the pruned tree
  tau <- Clone(tau.max)
  #the threshold for the pruning decision is calculated
  K <- cutoff*sqrt(nobs*log(nobs))
  #every leaf with delta<K=cutoff*sqrt(nlogn) is pruned
  Prune(tau, function(child) child$criticalCutoff>=K)
  return(tau)
}
```

R-Code 7.7: The pruneTauMax-function

pruneTauMax uses the offered Prune-function to prune $\tau_{\max}$. Since we want to save both, $\tau_{\max}$ and $\tilde{\tau}_t$, we first duplicate the tree. One needs to use Clone to do this: Initializing tau <- tau.max is not sufficient, doing so, both variable pointers still aim at the same tree object in the memory and after pruning tau and tau.max would both equal $\tilde{\tau}_t$. Next the threshold K $= C\sqrt{n \log n}$ is calculated and the tree is pruned by calling Prune. Prune takes a tree and a pruning function as input. A pruning function must be a function of a node outputting a boolean value. If the output is FALSE, the node is pruned. If it is TRUE, it is not. The pruning function used here calculates the censored time-sensitive pruning measure by calling calculatePruningMeasure and outputs FALSE, if the value is below the threshold K, TRUE otherwise.

7.3.1 The calculatePruningMeasure-function

```
calculatePruningMeasure <- function(child, measure) {
  #check if the chosen measure is the Kullback-Leibler divergence
  if (measure=="KLD") {
    #if the node checked is the root node, we return infinity:
    #the root node is the empty string "e", and "e" is never
    #pruned
    if(child$isRoot) {
      return(Inf)
    #if the node checked is not the root node...
    } else {
      #we calculate and return the pruning measure delta
      parent <- child$parent
      delta <- parent$counter*KLD(child$transitions,
                                  parent$transitions)
      return(delta)
    }
  #check if the chosen measure is the L1-norm
  } else if (measure=="L1") {
    if(child$isRoot) {
      return(Inf)
    } else {
      parent <- child$parent
      delta <- parent$counter*L1(child$transitions,
                                 parent$transitions)^2
      return(delta)
    }
  #if measure is not "KLD" or "L1" we throw an error
  } else {
    stop("measure must be KLD or L1.")
  }
}
```

R-Code 7.8: The calculatePruningMeasure-function

calculatePruningMeasure, cf. R-Code 7.8, basically is self-explaining. The censored time-sensitive pruning measure defined in Definition 5.5 is calculated if the cho-

sen measure is the Kullback-Leibler divergence, cf. Definition 4.1. The censored time-sensitive pruning measure defined in Corollary 5.10 is calculated if the L^1-norm was chosen. To calculate the Kullback-Leibler divergence and the L^1-norm, two auxiliary functions, KLD and L1, are used.

7.3.2 The KLD-function

```
KLD <- function(transitions.child, transitions.parent) {
  #initialize the result to be zero
  res <- 0
  sum.child <- sum(transitions.child)
  sum.parent <- sum(transitions.parent)
  #calculate and add up all summands
  for (i in 1:length(transitions.child)) {
    summand <- ifelse(transitions.child[i]/sum.child==0, 0,
                      (transitions.child[i]/sum.child)
                      *log(transitions.child[i]/sum.child
                      *sum.parent/transitions.parent[i]))
    res <- res + summand
  }
  return(unname(res))
}
```

R-Code 7.9: The KLD-function

7.3.3 The L1-function

```
L1 <- function(transitions.child, transitions.parent) {
  sum.child <- sum(transitions.child)
  sum.parent <- sum(transitions.parent)
  res <- sum(abs(transitions.child/sum.child-transitions.parent
                 /sum.parent))
  return(unname(res))
}
```

R-Code 7.10: The L1-function

7.4 The tuneC-function

The tuneC-function, cf. R-Code 7.11, implements the tuning procedure described in Chapter 6, i.e. it offers the tools to run Algorithm 5.6 multiple times for different cutoff values comfortably.
For each fit the values of the estimated log-likelihood function, cf. Chapter 6.1, and the model complexity, cf. Chapter 6.2, are calculated. The function outputs a table that contains these characteristics for each cutoff. This way AIC and BIC (as well as other information criteria based on these characteristics) can be easily computed. Via the optional tree.changes-argument, the user can specify whether the number of time steps where the shape of the estimated context tree changes should be computed additionally. Internally this number is computed by calling the treeChanges-function. The number of tree changes will play an important role in the later Chapter 11, where we are going to develop smoothing techniques for tiVLMC-models. Hence we postpone the discussion of the treeChanges-function and its dependencies to Chapter 11.
If the user has access to multiple CPUs, tuneC should be handed the input argument parallel=T. Then tuneC distributes the computational effort over all available CPU cores.

```
tuneC <- function(data, cutoffs, ties=NULL, from=NULL, to=NULL,
                  nmin=2, max.length=NULL, measure,
                  tree.changes=F, parallel=F) {
  #sort the cutoff values and omit duplicates
  cutoffs <- unique(sort(cutoffs))
  #fetch the smallest cutoff value
  C.min <- min(cutoffs)
  #if the smallest cutoff value is non-positive, throw an error
  if (C.min <= 0) {
    stop("The cutoff-vector contains non-positive elements.")
  }
  #the largest model is the one using C.min
  estimate1 <- estimateTau(data=data, ties=ties, from=from,
                           to=to, nmin=nmin,
                           max.length=max.length, cutoff=C.min,
                           measure=measure, log.like=T)
  #the tuning results will be saved here
```

```
#if tree.changes is true, the number of tree changes is
#computed for each fit
if (tree.changes) {
  result <- c(C=C.min,
              model.complexity=estimate1$model.complexity,
              log.like=estimate1$log.like,
              tree.changes=treeChanges(estimate1$tau))
} else {
  result <- c(C=C.min,
              model.complexity=estimate1$model.complexity,
              log.like=estimate1$log.like)
}
#now the models are fitted for the other values in cutoffs
#if parallel is false, a normal for-loop is used, else
#we use a foreach-loop and each of the available k cores
#computes (length(cutoffs)-1)/k iterations
tau1 <- estimate1$tau
if (isTRUE(parallel)) {
  result.temp <- foreach(C=cutoffs[2:length(cutoffs)],
                         .errorhandling="remove",
                         .packages=c("tictoc", "data.tree"),
                         .export=c("estimateTau", "buildTauMax",
                                   "initializeRootNode",
                                   "addBranch", "initializeNode",
                                   "updateNode", "pruneTauMax",
                                   "calculatePruningMeasure",
                                   "KLD", "L1",
                                   "getTransitionProbability",
                                   "calculateLogLikelihood",
                                   "calculateModelComplexity",
                                   "treeChanges",
                                   "identicalNodes"),
                         .combine=rbind) %dopar% {
    estimate <- estimateTau(data=data, ties=ties, from=from,
                            to=to, nmin=nmin,
                            max.length=max.length, cutoff=C,
                            measure=measure, log.like=T,
```

```
                              tau.max=tau1)
    if (tree.changes) {
     result.loop <- c(C=C,
                      model.complexity=estimate$model.complexity,
                      log.like=estimate$log.like,
                      tree.changes=treeChanges(estimate$tau))
    } else {
     result.loop <- c(C=C,
                      model.complexity=estimate$model.complexity,
                      log.like=estimate$log.like)
    }
    result.loop
  }
  result <- rbind(result, result.temp)
} else {
  for (C in cutoffs[2:length(cutoffs)]) {
    #tau.max is not depending on C, we only calculate it once
    #for a bigger C tau is included in the tau of a smaller C
    #thus we can replace tau.max by the tau of the smaller C
    estimate <- estimateTau(data=data, ties=ties, from=from,
                            to=to, nmin=nmin,
                            max.length=max.length, cutoff=C,
                            measure=measure, log.like=T,
                            tau.max=tau1)
    if (tree.changes) {
     result <- rbind(result, c(C=C,
                     model.complexity=estimate$model.complexity,
                     log.like=estimate$log.like,
                     tree.changes=treeChanges(estimate$tau)))
    } else {
     result <- rbind(result, c(C=C,
                     model.complexity=estimate$model.complexity,
                     log.like=estimate$log.like))
    }
  }
}
rownames(result) <- NULL
```

```
  return(result)
}
```

R-Code 7.11: The tuneC-function

Internally, the tuneC-function makes use of the fact that the tiVLMC-models are nested for decreasing values of the cutoff, recall (6.1) for this. Therefore it makes sense to compute the model fit using the smallest cutoff value first. This yields the largest model. Then we can continue by running the fitting algorithm again for the second largest cutoff and so on, but instead of calculating $\tau_{\max}$ again in step one of the algorithm and pruning it, we directly start with step two and prune the estimated context tree of the fit with the larger cutoff.

7.4.1 The calculateLogLikelihood-function

```
calculateLogLikelihood <- function(data, ties, tau) {
  #time consumption is measured and printed to the console
  tic(msg="likelihood computation finished")
  #we initialize the return
  LL <- 0
  #we run through every observation and...
  for (row in 1:dim(data)[1]) {
    #... fetch the frequency count of the observation
    freq <- ifelse(!is.null(ties), ties[row], 1)
    #at every time point...
    for (t in 1:(dim(data)[2]-1)) {
      #... we fetch the past states and the future state
      past <- data[row, 1:t]
      future <- data[row, t+1]
      #if the future is NA, this observation does not impact the
      #likelihood calculation
      if (!is.na(future)) {
        #we calculate the transition probability and...
        prob <- getTransitionProbability(past, future, tau[[t]])
        #... add the summand to the sum, while taking into respect
        #that this observation appeared freq times
```

```
        LL <- LL+freq*log(prob)
      }
    }
  }
  toc()
  return(LL)
}
```

R-Code 7.12: The calculateLogLikelihood-function

The calculateLogLikelihood-function, cf. R-Code 7.12, computes the estimated log-likelihood (6.2), also cf. Chapter 6.1, of the model fit.
It takes the training data data, the frequencies ties and the fitted context trees $\tilde{\tau}_t$ stored in the list tau as inputs. Then for each observation the likelihood of observing this observation in the fitted model is computed. These values are next logarithmized and added up which yields the estimated log-likelihood of the model fit. Within the computation the auxiliary function getTransitionProbability is used. This function extracts the transition probability of a given past w and future x, i.e. it extracts the context $c_t(w)$ and the transitions-attribute from the context tree $\tilde{\tau}_t = \text{tau[[t]]}$ and computes and outputs $\tilde{P}_t\left(x \,|\, c_t(w)\right)$.

7.4.2 The getTransitionProbability-function

```
getTransitionProbability <- function(past, future, tree) {
  #in the following we switch between child and parent nodes
  parent <- tree
  #if the most recent past is censored, the transition
  #probabilities are those of the empty string
  if (is.na(past[length(past)])) {
    #thus we fetch those transition counts...
    transitions <- parent$transitions
    #... and use them to calculate the transition probability
    trans.prob <- unname(transitions[future]/sum(transitions))
    return(trans.prob)
    #if the most recent past is not censored,
    #we still need to account for NAs in the past, i.e.
```

```
    #we need to extract the maximal possible past
  } else if (anyNA(past)) {
    past <- past[(max(which(is.na(past)))+1):length(past)]
  }
  transitions <- NULL
  for (i in length(past):1) {
    #we navigate down the branch of the tree given by past
    state <- past[i]
    child <- parent[[state]]
    #if not the whole past is contained as a branch, we will
    #encounter a missing child
    if (is.null(child)) {
      #then the context is the path from the root to the parent
      #of the missing child
      transitions <- parent$transitions
      break
    }
    parent <- child
  }
  #if we do not encounter a missing child, the whole past is
  #contained in a branch of the tree
  #in this case the context is the whole past
  if (is.null(transitions)) transitions <- child$transitions
  #we calculate the transition probability
  trans.prob <- unname(transitions[future]/sum(transitions))
  return(trans.prob)
}
```

R-Code 7.13: The getTransitionProbability-function

7.4.3 The calculateModelComplexity-function

```
calculateModelComplexity <- function(alphabet, tau) {
  #we calculate the size of the alphabet
  m <- length(alphabet)
  #we initialize the model complexity
```

```
  gamma <- 0
  #we run through all estimated trees and count the nodes
  for (i in 1:length(tau)) {
    #if the empty string has m children, e is not an element of
    #tau.i, else it is
    complete.root <- length(tau[[i]]$children) == m
    card <- ifelse(complete.root, tau[[i]]$totalCount-1,
                   tau[[i]]$totalCount)
    gamma <- gamma+card
  }
  gamma <- (m-1)*gamma
  return(gamma)
}
```

R-Code 7.14: The calculateModelComplexity-function

The calculateModelComplexity-function, cf. R-Code 7.14, computes and outputs the number of model parameters of the fitted tiVLMC.
It straightforwardly carries out the computations explained in Chapter 6.2. First the cardinality of the state space M is computed and then the nodes of the context trees $\tilde{\tau}_t$ stored in the list tau are counted.

8 Examples

8.1 Applying the algorithm

We are going to apply the programmed Algorithm 5.6 to different data sets: In the first data set the observations are uncensored. The second data set is subject to totally random, unstructured censoring. In the third data set an entry and an exit time are generated for each observation independently and independent from the observations, i.e. the observations are subject to independent left and right censoring. Last we are going to take a look at what will happen if the censoring depends on the observations, i.e. if the assumptions that imply the consistency of Algorithm 5.6, cf. Corollary 5.9, are violated.
We choose $T = 4$ and generate $n = 100{,}000$ observations from the tiVLMC X proposed in Example 3.11. The observations are saved in the data set data.
The first six observations are displayed in the following R-Code 8.1.

```
head(data)

##      [,1] [,2] [,3] [,4]
## [1,] "a"  "i"  "i+" "i"
## [2,] "a"  "a"  "a"  "i"
## [3,] "a"  "a"  "a"  "a"
## [4,] "a"  "i"  "i"  "i"
## [5,] "a"  "a"  "i+" "i"
## [6,] "a"  "a"  "a"  "d"
```

R-Code 8.1: The first six out of 100,000 observations of the tiVLMC X proposed in Example 3.11

8.1.1 No censoring

We infer the context trees τ_1, τ_2 and τ_3 by calling the estimateTau-function explained in Chapter 7.1. We use a cutoff value of cutoff=1, require a minimum occurrence counter of nmin=2 and prune the trees with the Kullback-Leibler divergence, i.e. measure="KLD".

```
#run the algorithm
estimate <- estimateTau(data=data, nmin=2, cutoff=1, measure="KLD")
#plot the estimates of tau.1, tau.2 and tau.3 and their
#non-pruned versions
plotTree(estimate$tau[[1]])
plotTree(estimate$tau.max[[1]])
plotTree(estimate$tau[[2]])
plotTree(estimate$tau.max[[2]])
plotTree(estimate$tau[[3]])
plotTree(estimate$tau.max[[3]])
```

R-Code 8.2: Inferring the context trees $\tilde{\tau}_1$, $\tilde{\tau}_2$ and $\tilde{\tau}_3$ from uncensored data

The last six lines of R-Code 8.2 call the plotTree-function (which itself depends on the recursiveColoring-function), cf. Codes 8.3 and 8.4, which specifies some cosmetic options and outputs the plots of $\tilde{\tau}_1$, $\tilde{\tau}_2$, $\tilde{\tau}_3$ and their non-pruned versions $\tau_{\max}$ at $t = 1, 2, 3$ respectively.

```
plotTree <- function(tree) {
  #finds the height of the input tree
  height <- tree$height
  #for each level of the tree we need a different grey
  colors <- grey.colors(n=height, start=1, end=0.6)
  #global settings that are applied to all levels of the tree
  SetGraphStyle(tree, rankdir="TB")
  SetEdgeStyle(tree, arrowhead="normal", penwidth=1, dir="back")
  SetNodeStyle(tree, style="filled", shape="box",
               fillcolor=colors[1], fontcolor="black",
               fontname="Time-Roman")
  #level-wise settings (coloring)
  children <- tree$children
  #recursive coloring of all children of a node
  recursiveColoring(children=children, colors=colors)
  plot(tree)
}
```

R-Code 8.3: The plotTree-function

```
recursiveColoring <- function(children, colors) {
  #for every node in children specify the color depending on
  #the level of the node, then move over to the children of
  #the node and recursively call this function again
  for (child in children) {
    SetNodeStyle(child, style="filled", shape="box",
                 fillcolor=colors[child$level],
                 fontcolor="black", fontname="Time-Roman")
    children <- child$children
    recursiveColoring(children=children, colors=colors)
  }
}
```

R-Code 8.4: The recursiveColoring-function

The plots are displayed in the following three Figures 8.1, 8.2 and 8.3.

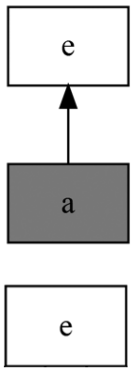

Figure 8.1: $\tau_{\max}$ at $t = 1$ (top) and its pruned version $\tilde{\tau}_1$ (bottom)

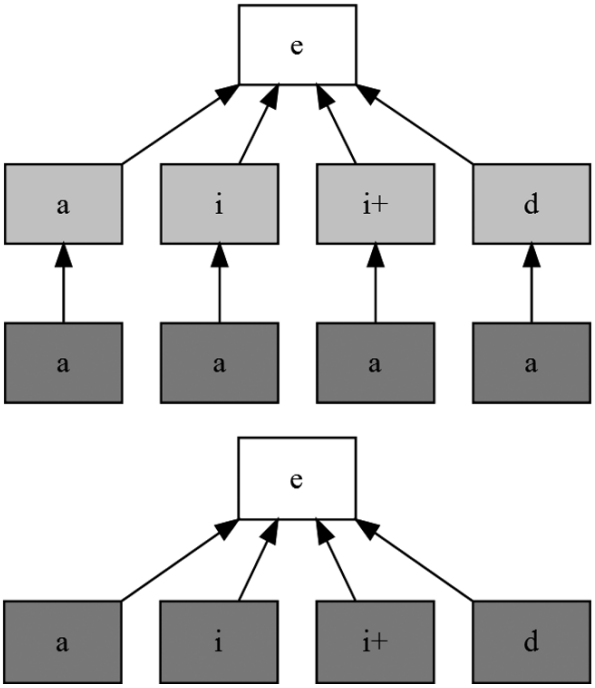

Figure 8.2: $\tau_{\max}$ at $t = 2$ (top) and its pruned version $\tilde{\tau}_2$ (bottom)

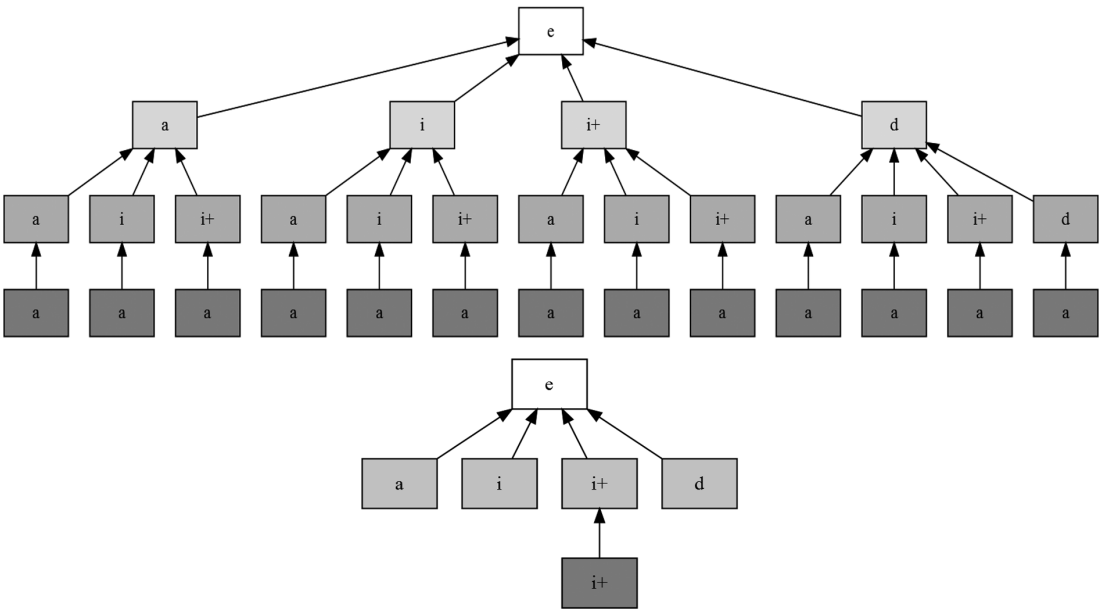

Figure 8.3: $\tau_{\max}$ at $t = 3$ (top) and its pruned version $\tilde{\tau}_3$ (bottom)

Comparing the inferred context trees with the true context trees, cf. Figure 3.1, one sees that here

$$\tilde{\tau} = \tau$$

holds and thus the algorithm found the true context tree.
Pruning with the L^1-norm, i.e. moving over to measure="L1", instead of the Kullback-Leibler divergence yields the same result.

8.1.2 Totally random censoring

Now we censor the data data with totally random censoring. By this we mean that every X_t^i, with $t = 1, 2, 3, 4$ and $i = 1, ..., 100{,}000$, is censored independently from each other with probability $p \in (0, 1)$. $p = 0$ gives the uncensored setting described in Chapter 8.1.1, if $p = 1$, there is nothing observable and assumption (H.5) would be violated. Here we choose $p = 10\%$. Thus the explicit distribution of the censoring variable J^i, $i = 1, ..., 100{,}000$, is

$$\mathbb{P}\left(J_{1:4}^i = (j_4, j_3, j_2, j_1)\right) = \prod_{t=1}^{4} \mathbb{P}\left(J_t^i = j_t\right) = \prod_{t=1}^{4} \left(0.1(1 - j_t) + 0.9 j_t\right)$$

with $(j_4, j_3, j_2, j_1) \in \{0, 1\}^4$. The assumptions (H.1)-(H.5) are fulfilled and, additionally, $J_1^i, ..., J_4^i$ are independent for every fixed i.
As explained in Chapter 7.1, the function estimateTau interprets NAs as censoring, i.e. we replace every X_t^i for which $J_t^i = 0$ holds by NA. The modified data set is called data.random.
The first six observations are displayed in R-Code 8.5.

```
head(data.random)

##      [,1] [,2] [,3] [,4]
## [1,] "a"  "i"  "i+" "i"
## [2,] "a"  "a"  "a"  "i"
## [3,] "a"  "a"  "a"  "a"
## [4,] NA   "i"  "i"  "i"
## [5,] NA   "a"  "i+" "i"
## [6,] "a"  "a"  "a"  "d"
```

R-Code 8.5: The first six out of 100,000 totally random censored observations of the tiVLMC X proposed in Example 3.11

With the same input options as in the uncensored setting, cf. Chapter 8.1.1, we calculate $\tilde{\tau}_1, \tilde{\tau}_2$ and $\tilde{\tau}_3$ by calling estimateTau.

```
#run the algorithm
estimate <- estimateTau(data=data.random, nmin=2, cutoff=1,
                        measure="KLD")
```

R-Code 8.6: Inferring the context trees $\tilde{\tau}_1$, $\tilde{\tau}_2$ and $\tilde{\tau}_3$ from totally random censored data

The output plots (by calling plotTree six times as in R-Code 8.2) are identical to the Figures 8.1, 8.2 and 8.3 and we can omit them here. Hence, as in the uncensored setting, the algorithm finds the correct context trees in the presence of totally random censoring, i.e.

$$\tilde{\tau} = \tau$$

holds.
This result again remains unchanged if the L^1-norm is used for pruning instead of the Kullback-Leibler divergence.

8.1.3 Independent left and right censoring

For each observation X^i an entry time t^i and an exit time T^i with $1 \leq t^i \leq T^i \leq 4$ are generated, $i = 1, ..., 100{,}000$. For the entry time t_i we choose the discrete distribution

$$\mathbb{P}\left(t^i = 1\right) = 0.7,\ \mathbb{P}\left(t^i = 2\right) = 0.25,\ \mathbb{P}\left(t^i = 3\right) = 0.05.$$

The exit time T^i depends on the entry time and is defined by

$$T^i := \max\{4, t^i + R - 1\},$$

where $R \sim \text{Bin}\left(4, \sqrt[4]{0.4}\right)$. Then we set

$$(J_4^i, J_3^i, J_2^i, J_1^i) := \left(\mathbb{1}\left(t^i \leq 4 \leq T^i\right), \mathbb{1}\left(t^i \leq 3 \leq T^i\right), \mathbb{1}\left(t^i \leq 2 \leq T^i\right), \mathbb{1}\left(t^i \leq 1 \leq T^i\right)\right).$$

If, for example, the entry time t^i is two and the exit time T^i is three, one will have $(J_4^i, J_3^i, J_2^i, J_1^i) = (0, 1, 1, 0)$.

Note that all the assumptions (H.1)-(H.5) are fulfilled, but contrary to the total random censoring setting in Chapter 8.1.2, $J_1^i, ..., J_4^i$ are not independent for each fixed i.
The modified data set is called data.independent.
The first six observations of this data set are displayed in R-Code 8.7.

```
head(data.independent)

##      [,1] [,2] [,3] [,4]
## [1,] "a"  "i"  "i+" "i"
## [2,] "a"  "a"  "a"  "i"
## [3,] "a"  "a"  "a"  "a"
## [4,] "a"  NA   NA   NA
## [5,] "a"  "a"  "i+" NA
## [6,] "a"  "a"  "a"  "d"
```

R-Code 8.7: The first six out of 100,000 independent left and right censored observations of the tiVLMC X proposed in Example 3.11

Using the same input options as in the two earlier scenarios, cf. Chapters 8.1.1 and 8.1.2, we calculate $\tilde{\tau}_1, \tilde{\tau}_2$ and $\tilde{\tau}_3$ with estimateTau:

```
#run the algorithm
estimate <- estimateTau(data=data.independent, nmin=2, cutoff=1,
                        measure="KLD")
```

R-Code 8.8: Inferring the context trees $\tilde{\tau}_1$, $\tilde{\tau}_2$ and $\tilde{\tau}_3$ from independent left and right censored data

We obtain the same result as before: The algorithm finds the correct context trees in the presence of independent left and right censoring, i.e.

$$\tilde{\tau} = \tau$$

holds under this type of censoring as well and hence we can skip the visual displaying of the estimators and instead refer to Figures 8.1, 8.2 and 8.3.
This result, as seen earlier, remains unchanged if the Kullback-Leibler divergence is exchanged with the L^1-norm.

8.1.4 Dependent censoring

Let $J_1^i, ..., J_4^i$ be independent with

$$\mathbb{P}^{J_1^i} = \mathbb{P}^{J_2^i} = \mathbb{P}^{J_3^i} = \delta_1$$

and

$$\mathbb{P}\left(J_4^i = 1\right) = \begin{cases} 1, & X_{2:3} \neq (i_+, i_+) \\ 0.4840, & X_{2:4} = (a, i_+, i_+) \\ 0.4909, & X_{2:4} = (i, i_+, i_+) \\ 0.0330, & X_{2:4} = (i_+, i_+, i_+) \\ 0.4818, & X_{2:4} = (d, i_+, i_+) \end{cases} \tag{8.1}$$

for $i = 1, ..., 100{,}000$. I.e. at the times $t = 1, 2, 3$ all observations are uncensored. Every observation that has been in state i_+ at $t = 2$ and $t = 3$ has a 95.15%, 40.91%, 96.70% or 31.82% chance of being censored at $t = 4$ depending on whether it is in state a, i, i_+ or d at $t = 4$ respectively.
Here assumption (H.3) is clearly violated: J_4^i depends on X_4^i for each fixed i. (H.1), (H.2), (H.4) and (H.5) hold.
The joint distribution of $J_{1:4}^i$ is the product distribution of $J_1^i, ..., J_4^i$.
Censoring the data set data in this way yields the new data set data.dependent.
The first six observations of data.dependent happen to be uncensored and hence can be looked at in R-Code 8.1.
Handing over the same input arguments as in the earlier examples, we estimate τ_1, τ_2 and τ_3 by running Algorithm 5.6.

```
#run the algorithm
estimate <- estimateTau(data=data.dependent, nmin=2, cutoff=1,
                        measure="KLD")
```

R-Code 8.9: Inferring the context trees $\tilde{\tau}_1$, $\tilde{\tau}_2$ and $\tilde{\tau}_3$ from dependently censored data

While the estimates $\tilde{\tau}_1$ and $\tilde{\tau}_2$ still equal the true context trees τ_1 at $t = 1$ and τ_2 at $t = 2$ respectively, the estimate $\tilde{\tau}_3$ differs from its true value τ_3. The following Figure 8.4 visualizes the pruning done by the algorithm.

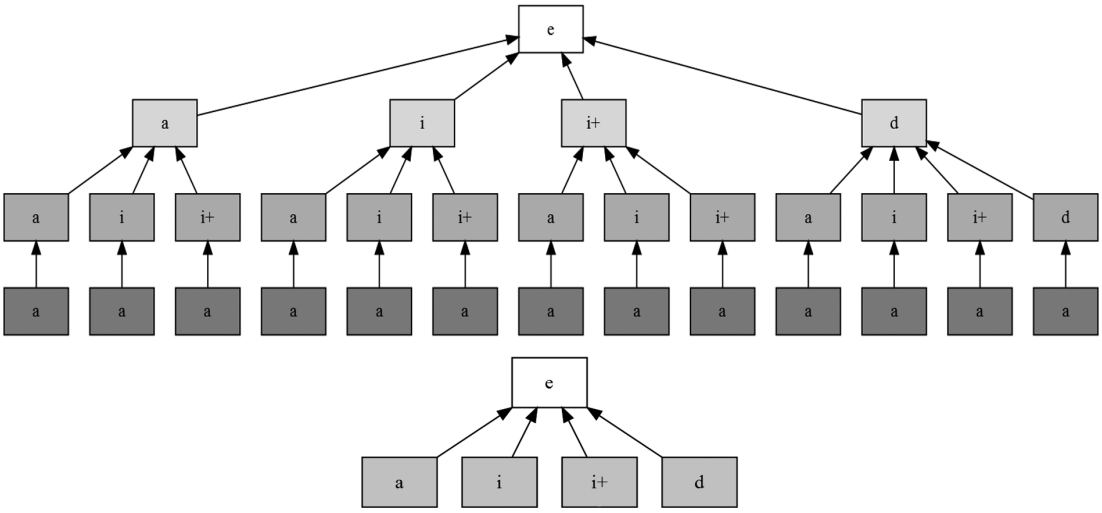

Figure 8.4: $\tau_{\max}$ at $t = 3$ (top) and its pruned version $\tilde{\tau}_3$ (bottom) in the presence of dependent censoring

The estimated context tree $\tilde{\tau}_3$ at $t = 3$ is missing the (i_+, i_+)-branch and thus indicates that

$$P_3\left(\cdot \,|\, (i_+, i_+)\right) = P_3(\cdot \mid i_+)$$

holds, which is false. The reason for this is the violation of assumption (H.3). In fact, the distribution of J_4^i, cf. (8.1), was precisely chosen to trigger this false pruning decision and thus illustrate the importance of the independence between the observation and its censoring variable stated in assumption (H.3). The limitations of the model explained in Chapter 5.4 are confirmed by this showcase.
To calculate $\mathbb{P}^{J_4^i}$ in a way that maximizes the chance of pruning the (i_+, i_+)-branch down to i_+, the censored time-sensitive pruning measure

$$\tilde{\Delta}_3((i_+, i_+)) = \tilde{N}_{3+}((i_+, i_+))\mathrm{D}\left[\tilde{P}_3\left(\cdot \,|\, (i_+, i_+)\right) || \tilde{P}_3\left(\cdot \,|\, i_+\right)\right]$$

must be minimized. To do so, we make the ansatz that we can express the censored counting variables at $t = 4$ by their uncensored versions minus the number

$$C_x \in \{0, ..., N_{t+1}((x, i_+, i_+))\}$$

of observations censored at $t = 4$, i.e.

$$\tilde{N}_4((x, i_+, i_+)) = N_4((x, i_+, i_+)) - C_x,$$

$x \in M$. This implies that

$$\begin{aligned}\tilde{N}_{3+}((i_+, i_+)) &= \sum_{x \in M} \tilde{N}_4((x, i_+, i_+)) \\ &= \sum_{x \in M} (N_4((x, i_+, i_+)) - C_x) \\ &= N_3((i_+, i_+)) - \sum_{x \in M} C_x,\end{aligned}$$

$$\tilde{N}_4((x, i_+)) = N_4((x, i_+)) - C_x$$

and

$$\begin{aligned}\tilde{N}_{3+}(i_+) &= \sum_{x \in M} \tilde{N}_4((x, i_+)) \\ &= \sum_{x \in M} (N_4((x, i_+)) - C_x) \\ &= N_4(i_+) - \sum_{x \in M} C_x.\end{aligned}$$

Thus the censored empirical measures are given by

$$\tilde{P}_3\left(x \,|\, (i_+, i_+)\right) = \frac{\tilde{N}_4((x, i_+, i_+))}{\tilde{N}_{3+}((i_+, i_+))} = \frac{N_4((x, i_+, i_+)) - C_x}{N_3((i_+, i_+)) - \sum\limits_{x \in M} C_x}$$

and

$$\tilde{P}_3\left(x \,|\, i_+\right) = \frac{\tilde{N}_4((x, i_+))}{\tilde{N}_{3+}(i_+)} = \frac{N_4((x, i_+)) - C_x}{N_3(i_+) - \sum\limits_{x \in M} C_x},$$

$x \in M$. In this explicit case we have

$$\begin{aligned}N_4((a, i_+, i_+)) &= 62, \\ N_4((i, i_+, i_+)) &= 22, \\ N_4((i_+, i_+, i_+)) &= 91, \\ N_4((d, i_+, i_+)) &= 66, \\ N_4((a, i_+)) &= 423, \\ N_4((i, i_+)) &= 1{,}525, \\ N_4((i_+, i_+)) &= 423, \\ N_4((d, i_+)) &= 5{,}253,\end{aligned}$$

and thus

$$N_3((i_+, i_+)) = 241,$$
$$N_3(i_+) = 7{,}624,$$

cf. the following R-Code 8.10.

```
estimate <- estimateTau(data=data, nmin=2, cutoff=1, measure="KLD")
print(estimate$tau.max[[3]]$"i+"$"i+"$transitions)

##  a  i i+  d
## 62 22 91 66

print(estimate$tau.max[[3]]$"i+"$transitions)

##    a    i   i+    d
##  423 1525  423 5253

print(estimate$tau.max[[3]]$"i+"$"i+"$counter)

## [1] 241

print(estimate$tau.max[[3]]$"i+"$counter)

## [1] 7624
```

R-Code 8.10: Accessing the transitions- and counter-attribute of the node

Then we explicitly solve

$$\left(C_a^*, C_i^*, C_{i+}^*, C_d^*\right) = \underset{C_a, C_i, C_{i+}, C_d}{\arg\min}\ \tilde{\Delta}_3((i_+, i_+)) \tag{8.2}$$

subject to the $|M| = m$ constraints that $0 \leq C_x \leq N_{t+1}((x, i_+, i_+))$ has to hold for $x \in M$ by running through each of the

$$(62 + 1)(22 + 1)(91 + 1)(66 + 1) = 8{,}931{,}636$$

combinations of vectors (C_a, C_i, C_{i+}, C_d).
This computation is performed within 14 minutes of time on the CARL High-Performance Computing cluster of the University of Oldenburg, cf. Harfst (2020),

using 63 cores clocked at 2.2GHz of Intel Xeon (E5-2650 v4) CPUs. For parallelized code execution the doMPI-, cf. Weston (2013), and the foreach-package, cf. Analytics and Weston (2015), are used.

```
#initialize the CPU cluster
library(doMPI)
library(foreach)
cluster <- startMPIcluster()
registerDoMPI(cluster)
#read in the values of the counting variables
trans.child <- c(62, 22, 91, 66)
trans.parent <- c(423, 1525, 423, 5253)
#calculate the censored time sensitive pruning measure for each
#combination
result <- foreach(C.a=0:trans.child[1], .combine=rbind) %dopar% {
  result.core <- NULL
  for (C.i in 0:trans.child[2]) {
    for (C.iplus in 0:trans.child[3]) {
      for (C.d in 0:trans.child[4]) {
        C <- c(C.a, C.i, C.iplus, C.d)
        kld <- KLD(trans.child-C, trans.parent-C)
        result.core <- rbind(result.core, c(C, kld,
                             241-sum(C), (241-sum(C))*kld))
      }
    }
  }
  result.core
}
colnames(result) <- c("C.a", "C.i", "C.iplus", "C.d", "D", "N",
                      "Delta")
#shut down the CPU cluster
closeCluster(cluster)
mpi.quit()
#printing the row which contains the minimal value
result[which.min(result[,5]), ]
```

R-Code 8.11: Solving the minimization problem stated in equation (8.2)

The last line of R-Code 8.11 prints the solution

$$\left(C_a^*, C_i^*, C_{i+}^*, C_d^*\right) = (59, 9, 88, 21)$$

to the console. For this combination the censored time-sensitive pruning measure $\tilde{\Delta}_3((i_+, i_+))$ is minimal, taking a value of only 0.0052 (rounded to four decimal digits). In comparison, the cutoff that $\tilde{\Delta}_3((i_+, i_+))$ has to fall below for pruning, is

$$C\sqrt{n \log n} = 1\sqrt{100{,}000 \log(100{,}000)} \approx 1{,}072.98$$

(rounded to two decimal digits). If we now censor

$$\left(\frac{C_x^*}{N_4((x, i_+, i_+))} \cdot 100\right)\%$$

of the observations that are in state i_+ at $t = 2$ and $t = 3$ and in state x at $t = 4$, we get that the empirical measures $\tilde{P}_3\left(x \,|\, (i_+, i_+)\right)$ and $\tilde{P}_3\left(x \,|\, i_+\right)$ are as close to each other as possible, where closeness is measured by the censored time-sensitive pruning measure. Since

$$\begin{aligned}
\frac{C_a^*}{N_4((a, i_+, i_+))} &= \frac{59}{62} = 95.16\%, \\
\frac{C_i^*}{N_4((i, i_+, i_+))} &= \frac{9}{22} = 40.91\%, \\
\frac{C_{i_+}^*}{N_4((i_+, i_+, i_+))} &= \frac{88}{91} = 96.70\%, \\
\frac{C_d^*}{N_4((d, i_+, i_+))} &= \frac{21}{66} = 31.82\%,
\end{aligned}$$

the optimality of the choice of $\mathbb{P}^{J_4^i}$ is explained.

As stated earlier, this chapter highlights the fact that it is not possible to infer the true context tree if there are any unknown "bad" dependencies between the observations X^i and their censoring variables J^i, i.e. this chapter shows the importance of assumption (H.3) for the consistency of Algorithm 5.6, cf. Corollary 5.9. But, as stated at the end of Chapter 5.4, in practice only knowing the context tree under censoring should be sufficient for many purposes.

8.2 Tuning the algorithm

We will exemplarily demonstrate the tuning procedure explained in Chapter 6 by tuning Algorithm 5.6 applied to the data sets data and data.independent, cf. Chapters 8.1.1 and 8.1.3 respectively.

8.2.1 No censoring

We run Algorithm 5.6 on the uncensored data set data generated in Chapter 8.1. As earlier we use $n_{\min} = \text{nmin} = 2$ and choose the Kullback-Leiber divergence as the measure, i.e. measure=KLD. For 22,300 values of the cutoff

$$C = \text{cutoff} \in \mathscr{C} = \{0.0001, 0.0002, ..., 2, 2.01, 2.02, ..., 25\}$$

tiVLMC-models are fitted. Here we use the tuneC-function displayed and explained in Chapter 7.4 and R-Code 7.11. For each fitted model tuneC calculates the estimated log-likelihood, cf. Chapter 6.1, and the model complexity, cf. Chapter 6.2. We use these values to calculate and save AIC and BIC. The following R-Code 8.12 shows the implementation.

```
#defining the cutoff lattice
C <- c(seq(from=0.0001, to=2, by=0.0001), seq(from=2.01, to=25,
                                              by=0.01))
#tune the model
tuning <- tuneC(data=data, cutoffs=C, measure="KLD", parallel=T)
#calculate AIC and BIC
tuning <- cbind(tuning, 0, 0)
colnames(tuning)[4:6] <- c("AIC", "BIC")
tuning[, "AIC"] <- -2*tuning[, "log.like"]
                   +2*tuning[, "model.complexity"]
tuning[, "BIC"] <- -2*tuning[, "log.like"]
                   +log(100000)*tuning[, "model.complexity"]
#find the cutoffs that minimize AIC or BIC
C.AIC <- tuning[which.min(tuning[, "AIC"]), "C"]
C.BIC <- tuning[which.min(tuning[, "BIC"]), "C"]
```

R-Code 8.12: Tuning Algorithm 5.6 with the tuneC-function

The computations are carried out within six minutes on the CARL High-Performance Computing cluster, cf. Harfst (2020), using 250 2.2GHz cores of Intel Xeon (E5-2650 v4) CPUs.
The AIC and BIC values are plotted in Figure 8.5 and 8.7 respectively. Both figures consist of two plots: While the range of the first plot is $[0, 25]$ and thus includes the whole set $\mathscr{C}$, the bottom plot zooms into the interesting interval $[0, 1]$. The values of the information criterion are plotted as a continuous blue line. The vertical grey bars indicate the sampling rate: Within each segment the information criterion was computed for 1,000 equidistant values of C.
The scales-package, cf. Wickham (2016), was used for generating the figures.

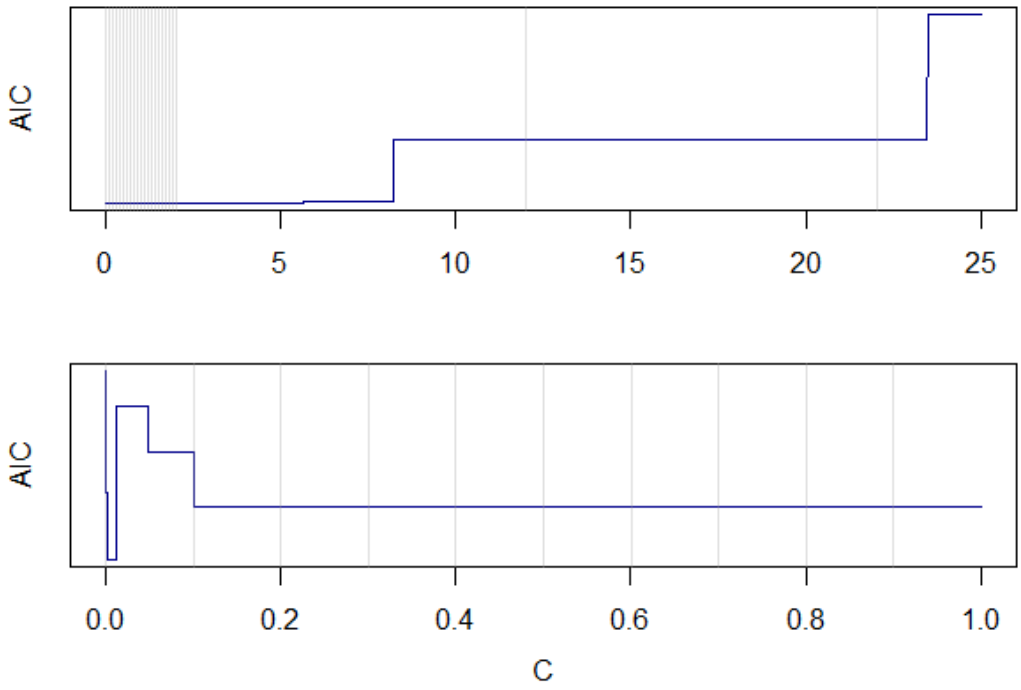

Figure 8.5: Tuning Algorithm 5.6 on the data data set with AIC

AIC is minimal for $C \in [0.0015, 0.0117]$. For all those cutoff values the same model estimate is output by the algorithm and here we arbitrarily choose $C_{\text{AIC}} = 0.0015$. The complexity of the corresponding model is 42 and the estimated log-likelihood is -245,878.00, which gives the minimal AIC of 491,840.00. The context tree estimates are displayed in the following Figure 8.6.

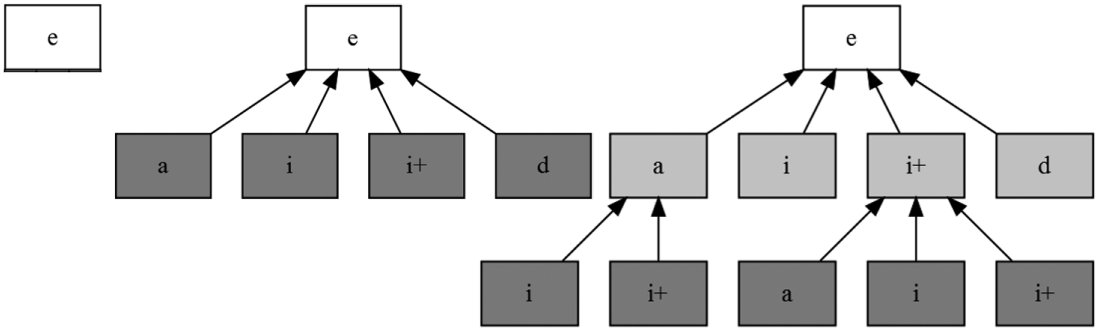

Figure 8.6: The AIC-tuned estimates $\tilde{\tau}_1$, $\tilde{\tau}_2$, $\tilde{\tau}_3$ (left to right)

The AIC-tuned model correctly estimates the first two context trees, i.e. $\tau_1 = \tilde{\tau}_1$ and $\tau_2 = \tilde{\tau}_2$ hold. However, it overfits the third context tree: $\tau_3 \preccurlyeq \tilde{\tau}_3$, $\tau_3 \neq \tilde{\tau}_3$.

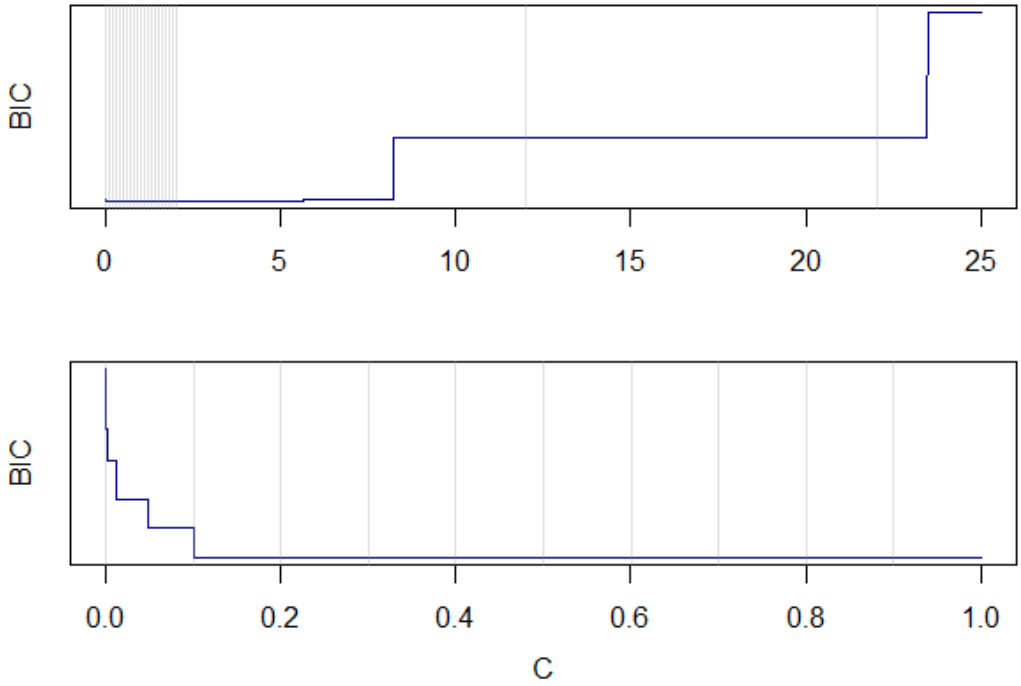

Figure 8.7: Tuning Algorithm 5.6 on the data data set with BIC

BIC on the other hand is minimal for $C \in [0.1004, 5.64]$. Again, all those cutoff values lead to the same model estimate and we choose $C_{\text{BIC}} = 1$. The corresponding model was already fitted and discussed in Chapter 8.1.1 and is known to be the true model. It has a complexity of 30 and the estimated log-likelihood is -245,892.40, yielding a minimal BIC-value of 495,284.70. For a plot of the context tree estimates go back to Figures 8.1-8.3. Contrary to the AIC, here BIC detects the correct tiVLMC-model.

8.2.2 Independent left and right censoring

Next we tune Algorithm 5.6 on the data set data.independent, which is subject to left and right censoring. The data was generated and the censoring distribution was given in Chapter 8.1.3. We continue with the same options and fit tiVLMC-models for the same 22,300 values of the cutoff

$$C = \text{cutoff} \in \mathscr{C} = \{0.0001, 0.0002, ..., 2, 2.01, 2.02, ..., 25\}.$$

Again we calculate AIC and BIC. This is analogous to R-Code 8.12, only now data.independent instead of data is used.
We kept the computational environment unchanged and the computation again took six minutes.

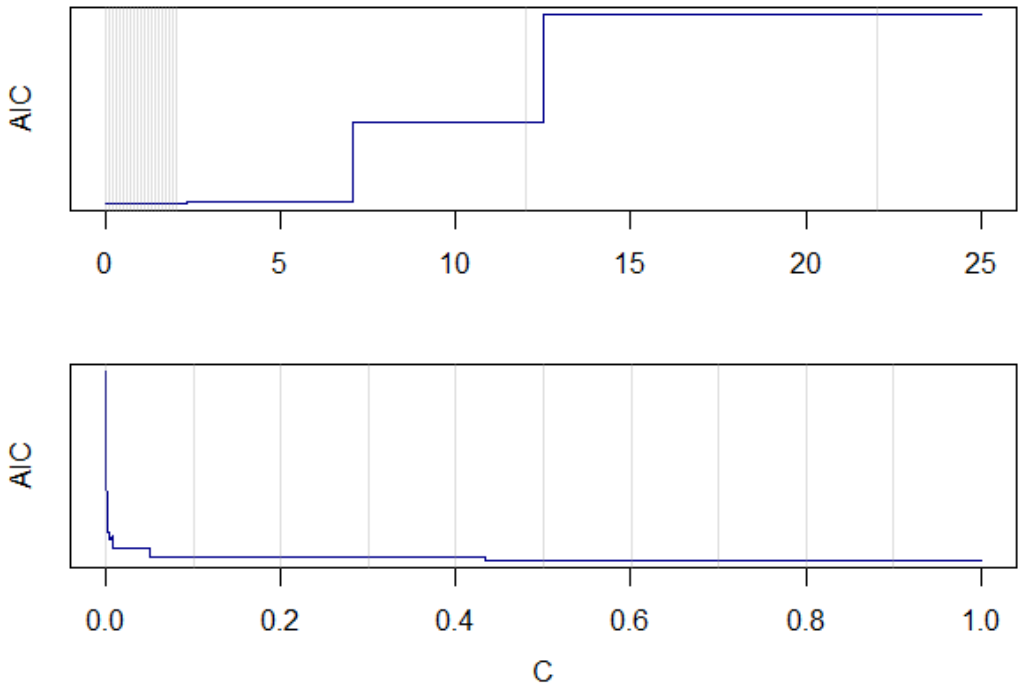

Figure 8.8: Tuning Algorithm 5.6 on the data.independent data set with AIC

In this setting with censoring present AIC is minimized if $C \in [0.4332, 2.32]$. Since this interval contains $C = 1$, the estimated context trees were already discussed in Chapter 8.1.3 and are known to be the true context trees. The model complexity (obviously) remains at 30 and the estimated log-likelihood decreases to -194,607.60, which gives a minimal AIC-value of 389,275.20.
Here it is interesting that the censoring seems to mask the dip in the AIC which

leads to the overfitting model in the uncensored case.

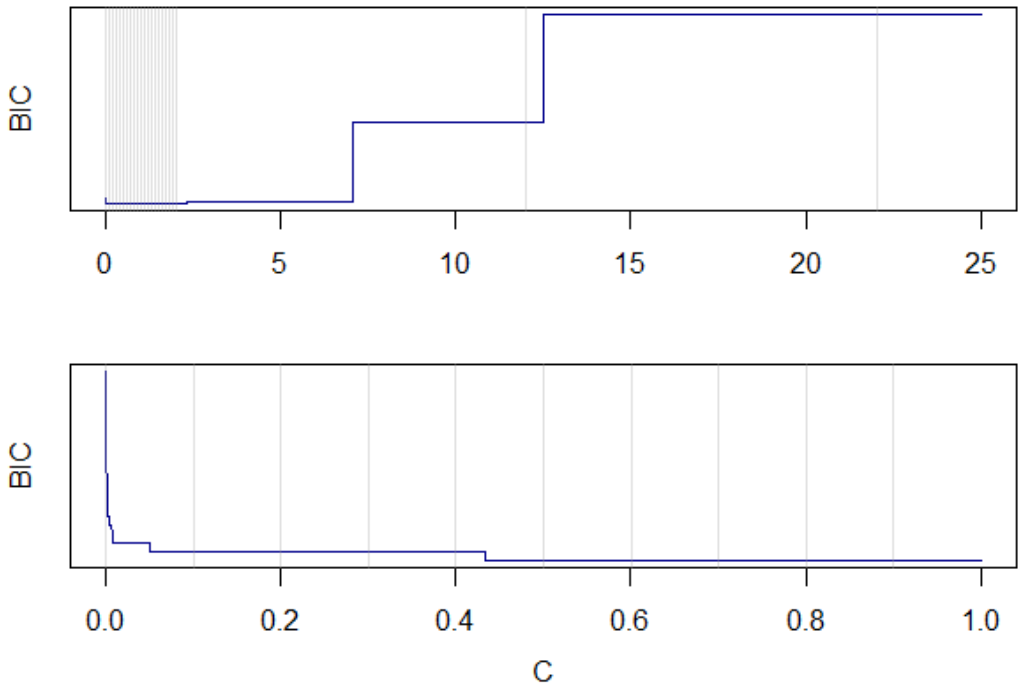

Figure 8.9: Tuning Algorithm 5.6 on the data.independent data set with BIC

On the other hand BIC is minimal for $C \in [0.4332, 2.32]$ and the results do not change in comparison to the uncensored setting in Chapter 8.2.1. The tuned model has a complexity of 30 and the estimated log-likelihood is -194,607.60, yielding a minimal BIC-value of 389,560.40.

9 The prospective reserve

Within this chapter we are going to discuss one of the most important quantities in insurance: the prospective reserve. The prospective reserve, also called "mathematical reserve", e.g. cf. Koller (2012), or "benefit reserve", e.g. cf. Bowers et al. (1997), is the monetary amount the insurance company has to hold onto in order to meet the expected monetary future liabilities it has towards the insurance holders. We develop a computation formula and implement the calculation in R.

9.1 A computation formula

Conventionally, payments between the insurance holder and the insurance company are distinguished for their direction: A payment initiated by the insurance holder, e.g. an annual premium, can be identified by its negative sign. On the contrary a payment flowing from the insurance company to the customer, e.g. a disability pension, is marked by a positive sign. Furthermore one can verbally label payments by their major characteristic: The payments that get triggered because the insured remains in his current state, e.g. she or he remains disabled, are called "sojourn payments". On the other hand payments that are triggered when the insured changes her or his current state are called "transition payments".
We give a mathematical definition and continue by looking at some examples:

Definition 9.1. Let $1 \leq t \leq T$ and $x_{1:t} \in M^{1:t}$. We denote the payment that gets triggered when the insured moves through $x_{1:t}$ up to t, i.e. the payment gets triggered if $X_{1:t} = x_{1:t}$ holds, by

$$b_t\left(x_{1:t}\right).$$

Payments flowing from the insurance holder to the insurance company are marked with a negative sign. Payments flowing from the insurance company to the insurance holder are marked with a positive sign.

For the ease of writing we also introduce a shorter notation that has similarities to the concept of context functions:

Definition 9.2. Let $1 \leq t \leq T$ and $x_{1:t} \in M^{1:t}$. Define

$$l_t^b := \min\Big\{1 \leq k \leq t : b_t\left(x_{t-k+1:t} \cdot x_{1:t-k}\right) = b_t\left(x_{t-k+1:t} \cdot x'_{1:t-k}\right) \text{ for all } x'_{1:t-k} \in M^{t-k}\Big\}.$$

We also use the shorter notation

$$b_t\left(x_{t-l_t^p+1:t}\right) := b_t\left(x_{1:t}\right)$$

and call

$$|b_t| := \max_{w \in M^{1:t}} l_t^p(w)$$

the order of the payments at t.

By definition the order $|b_t|$ of the payments at t is the minimal number of most recent states we need to know in order to check all payments for their triggering event. Similar to a transition probability that only needs the context and not the full past as its condition, the payments $b_t(x_{1:t})$ viewed as a function of the full past $x_{1:t}$ really only depend on $x_{t-|b_t|+1:t}$.
In the following let t_s denote the age of the insured at contract closing and t_e the age of the insured at maturity, $1 \leq t_s < t_e \leq T$. We assume that for $t > t_e$ and for $t < t_s$

$$b_t(x_{1:t}) = 0 \tag{9.1}$$

holds for every $x_{1:t} \in M^t$, i.e. no (non-zero) payments are made after the contract has matured or before the contract has started.

Example 9.3. Let $M = \{a, i, d\}$ where, as in the earlier Example 3.2, a again stands for active, i for invalid and d for deceased.

(a) The insurance company pays the bereaved of the insurance holder a death cover $D > 0$ if the insured dies before the maturity t_e (transition payment):

$$b_t\left((d, a)\right) = b_t\left((d, i)\right) = D$$

for $t_s + 1 \leq t \leq t_e$.

(b) After a waiting period of five time periods, the insurance pays out a disability annuity of size $A > 0$ at every time period to every invalid insurance holder under contract (sojourn payment):

$$b_t(i) = A$$

for $t_s + 5 \leq t \leq t_e$.

(c) The insurance holder and the insurance agree on an initial premium $P > 0$ that has to be paid for every time period in which the insurance holder is active and that increases at an annual rate of 10% (sojourn payment). The only exception to this is (d), i.e. when the insurance holder has been active for the last five consecutive years. I.e.

$$b_t(a \cdot w) = -P \cdot 1.1^{t-1}$$

for $w \neq (a, a, a, a)$ and $t_s \leq t \leq t_e$.

(d) Insurance holders that have been active for the last five consecutive years get paid out a health dividend $H > 0$. Since they are active they still have to pay the premium specified in (c). I.e.

$$b_t((a, a, a, a, a)) = H - P \cdot 1.1^{t-1}$$

for $t_s + 4 \leq t \leq t_e$.

In a world with non-zero interest, payments of the same monetary size made at different time points are of different value. Intuitively, if the risk-free interest rate is positive, receiving one monetary unit now is more valuable to the recipient than receiving one monetary unit at a later point in time. The earlier the money is received, the earlier it earns the recipient interest. The present value of the earlier payment is higher than the present value of the later payment. Mathematically we can formalize this by introducing the discounting factor:

Definition 9.4. For $1 \leq t \leq k \leq T$ let

$$v_{t,k}$$

be the present value at time t of a payment of one monetary unit at time k. We call $v_{t,k}$ the discounting factor.

Via the relation

$$r_t = \frac{1}{v_{t-1,\,t}} - 1. \tag{9.2}$$

the one-step discounting factor $v_{t-1,\,t}$ can be used to compute the risk-free interest rate r_t at t and vice versa.

Example 9.5. Assume that the risk-free interest rate is $r_1 = 0\%$ and $r_t = 2\%$ for all $2 \leq t \leq T$. Then we have $v_{1,\,1} = 1$ and $v_{t-1,\,t} = 0.980392$ for $2 \leq t \leq T$. The present value of receiving a payment of one monetary unit at t at time $t-1$ is

$$1 \cdot v_{t-1,\,t} = 1 \cdot 0.980392 = 0.980392.$$

On the other hand, if one deposits 0.980392 at time $t-1$ at the risk-free interest rate of $r_t = 2\%$, one ends up with exactly

$$0.980392 \cdot (1 + r_t) = 0.980392 \cdot 1.02 = 1$$

monetary units at t.

Making use of the Definitions 9.1 and 9.4, we can continue:

Definition 9.6. Let $1 \leq t \leq t_e$ and $x_{1:t_e} \in M^{t_e}$. The discounted future payments at time t of an insured that moved through $x_{1:t_e}$ are

$$B_t\left(x_{1:t_e}\right) := \sum_{k=t}^{t_e} v_{t,\,k}\, b_k\left(x_{1:k}\right).$$

Note that by (9.1)

$$B_t\left(x_{1:T}\right) = B_t\left(x_{t_e+1:T} \cdot x_{1:t_e}\right) = B_t\left(x_{1:t_e}\right)$$

holds for all $x_{t_e+1:T} \in M^{T-t_e}$.
$B_t\left(x_{1:t_e}\right)$ is the monetary amount that the insurance company needs at the beginning of time period t (i.e. before the payment $b_t\left(x_{1:t}\right)$ is made) to exactly meet all the future liabilities it has towards a customer that moves through $x_{1:t_e}$. In general $B_t\left(x_{1:t_e}\right)$ is unknown at time t since it depends on the future development of the insurance holder, i.e. on $x_{t+1:t_e}$. Thus the insurance company has to estimate

$B_t\left(x_{1:t_e}\right)$ somehow. As usual this is done by moving over to the expected value which one conditions on all information available up to the present time point t. If the insurance company sells the contract to a large number of customers, the strong law of large numbers A.1 will justify this estimation.

Definition 9.7. Let $1 \leq t \leq t_e$ and $x_{1:t} \in \mathrm{tpsupp}_t(X)$. We call

$$V_t\left(x_{1:t}\right) := \mathbb{E}\left[B_t\left(X_{1:t_e}\right) \mid X_{1:t} = x_{1:t}\right]$$

the pathwise prospective reserve of an insured that moved through $x_{1:t}$.

The pathwise prospective reserve of an insured that moved through $x_{1:t}$ can be calculated as stated in the following Proposition 9.8.

Proposition 9.8. *Let $1 \leq t \leq t_e$ and $x_{1:t} \in \mathrm{tpsupp}_t(X)$. It holds that*

$$V_t\left(x_{1:t}\right) = \sum_{x_{t+1:t_e} \in M^{t_e - t}} \mathbb{P}\left(X_{t+1:t_e} = x_{t+1:t_e} \mid X_{1:t} = x_{1:t}\right) \sum_{k=t}^{t_e} v_{t,\,k}\, b_k\left(x_{1:k}\right).$$

Recall that $|M^0| = |\{e\}| = 1$. We establish the convention that $X_{l:k} := e$ for $l > k$. In this way $V_{t_e}(x_{1:t_e})$ is well-defined. In the following we will also use that products ranging over empty sets are defined as one.

Proof of Proposition 9.8. First we write out the expectation:

$$\begin{aligned} V_t\left(x_{1:t}\right) &= \mathbb{E}\left[B_t\left(X_{1:t_e}\right) \mid X_{1:t} = x_{1:t}\right] \\ &= \sum_{x'_{1:t_e} \in M^{t_e}} \mathbb{P}\left(X_{1:t_e} = x'_{1:t_e} \,\middle|\, X_{1:t} = x_{1:t}\right) B_t\left(x'_{1:t_e}\right). \end{aligned}$$

Now if $x'_{1:t} \neq x_{1:t}$, we have

$$\mathbb{P}\left(X_{1:t_e} = x'_{1:t_e} \,\middle|\, X_{1:t} = x_{1:t}\right) = 0$$

and thus we obtain:

$$V_t(x_{1:t}) = \sum_{x_{t+1:t_e} \in M^{t_e-t}} \mathbb{P}(X_{t+1:t_e} = x_{t+1:t_e} \mid X_{1:t} = x_{1:t}) B_t(x_{1:t_e})$$
$$= \sum_{x_{t+1:t_e} \in M^{t_e-t}} \mathbb{P}(X_{t+1:t_e} = x_{t+1:t_e} \mid X_{1:t} = x_{1:t}) \sum_{k=t}^{t_e} v_{t,k} b_k(x_{1:k}).$$

□

In the classical Markov-setup the Markov assumption implies that the discounted future payments and the prospective reserve only depend on the present and not on the past. Therefore one only needs to consider solo-state prospective reserves $V_t(x)$ with $x \in M$ instead of the pathwise prospective reserves defined in Definition 9.7. Additionally, in the Markovian case the Chapman-Kolmogorov equations A.2 can be applied. These strong tools allow us to decompose every occurring probability into (sums and products of) one-step transition probabilities. This is how the famous Thiele equation follows. The Thiele equation allows for an efficient calculation of the prospective reserve in a recursive way. It was discovered by the Danish mathematician Thorvald Nicolai Thiele and first published in Gram (1910). Within the environment of continuous stochastic processes the Thiele equation takes the form of a differential equation.
Since we can neither apply the Markov assumption nor the Chapman-Kolmogorov equation A.2, we cannot copy the explained strategy. However, the probability in Proposition 9.8 can still be further decomposed, as we have already done multiple times:

$$\mathbb{P}(X_{t+1:t_e} = x_{t+1:t_e} \mid X_{1:t} = x_{1:t}) = \prod_{l=t}^{t_e-1} P_l(x_{l+1} \mid x_{1:l}). \tag{9.3}$$

Now, as we have learned earlier on in Chapter 3, if X is a tiVLMC with context function c, the set of transition probabilities $\mathbb{P}_c$ and the marginal distribution uniquely determine the tiVLMC, recall Definition 3.8 and Proposition 3.9. We have also learned that, as a direct consequence, one never needs to consider the complete past $x_{1:l}$ at time l. Instead it is sufficient to focus on the context $c_l(x_{1:l})$. Thus the following Corollary 9.9 is immediately implied.

Corollary 9.9. *Let* $1 \leq t \leq t_e$ *and* $x_{1:t} \in \mathrm{tpsupp}_t(X)$*. It holds that*

$$V_t(x_{1:t}) = \sum_{x_{t+1:t_e} \in M^{t_e-t}} \prod_{l=t}^{t_e-1} P_l(x_{l+1} \mid c_l(x_{1:l})) \sum_{k=t}^{t_e} v_{t,k}\, b_k(x_{1:k}).$$

In general $c_l(x_{1:l})$ can be expected to be of much shorter length than $x_{1:l}$ itself. Therefore the calculation of $V_t(x_{1:t})$ can be highly accelerated by using above computation formula instead of the formula stated in Proposition 9.8 combined with (9.3).

To summarize, in general neither the transition probabilities at t nor the payments at t depend on the complete past $x_{1:t}$. Recall Definition 3.3 of the order $|c_t|$ of a context function at t and the Definition 9.2 of the order $|b_t|$ of the payments at t: The transition probabilities at t viewed as functions of $x_{1:t}$ only depend on $x_{t-|c_t|+1:t}$ and the payments at t viewed as functions of $x_{1:t}$ only depend on $x_{t-|b_t|+1:t}$. Thus

$$l_t^{\min} := \max\{|b_t|, |c_t|\}$$

is the minimal length such that

$$b_t(x_{1:t}) = b_t\left(x_{t-l_t^{\min}+1:t}\right) \tag{9.4}$$

and

$$P_t(x_{t+1} \mid c_t(x_{1:t})) = P_t\left(x_{t+1} \,\middle|\, c_t\left(x_{t-l_t^{\min}+1:t}\right)\right) \tag{9.5}$$

hold simultaneously for all $x_{1:t} \in \mathrm{tpsupp}_t(X)$. Let

$$l^{\min} := \max\left\{\max_{t \leq k \leq t_e-1}\left\{l_t^{\min} + t - k\right\}, |b_{t_e}| + t - t_e\right\}.$$

Then in general $V_t(x_{1:t})$ viewed as a function of $x_{1:t}$ does not really depend on the complete past $x_{1:t}$ but only on $x_{t-l^{\min}+1:t}$, since $l^{\min}$ is chosen such that (9.4) and (9.5) are true for all the appearing context and payment functions in Corollary 9.9:

$$b_k\left(x_{t-|x|+1:k}\right) = b_k(x_{1:k})$$

holds for $k = t, ..., t_e$ if and only if

$$\left|x_{t-|x|+1:k}\right| = \left|x_{t-|x|+1:t}\right| + |x_{t+1:k}| = |x| + t - k \geq |b_k|$$

for $k = t, ...t_e$. And

$$c_k\left(x_{t-|x|+1:k}\right) = c_k\left(x_{1:k}\right)$$

holds for $k = t, ..., t_e - 1$ if and only if

$$\left|x_{t-|x|+1:k}\right| = |x| + t - k \geq |c_k|$$

for $k = t, ...t_e$.
At first glance the choice of $l^{\min}$ might seem complex, but it is really just the minimal length of the past such that the length of the state change sequences input into the payment and context functions are greater than or equal to the height of the payment and context trees. If one prefers a simpler, less sharp but still sufficient condition: Any past with a length greater than or equal to the maximum tree height occurring will work.
Therefore we can generalize Definition 9.7:

Definition 9.10. Let $t_s \leq t \leq t_e$ and $x_{t-|x|+1:t} \in M^{|x|}$ be a state sequence of length

$$t \geq |x| \geq l^{\min}$$

with $\mathbb{P}\left(X_{t-|x|+1:t} = x_{t-|x|+1:t}\right) > 0$. We call

$$V_t\left(x_{t-|x|+1:t}\right) := \sum_{x_{t+1:t_e} \in M^{t_e-t}} \prod_{l=t}^{t_e-1} P_l\left(x_{l+1} \,\middle|\, c_l\left(x_{t-|x|+1:l}\right)\right) \sum_{k=t}^{t_e} v_{t,k}\, b_k\left(x_{t-|x|+1:k}\right)$$

the pathwise prospective reserve of an insured that moved through $x_{t-|x|+1:t}$.

By Corollary 9.9 and the definition of $l^{\min}$ it holds that

$$V_t(x_{t-|x|+1:t}) = V_t(x_{t-|x|+1:t} \cdot x_{1:t-|x|}) = V_t(x_{1:t})$$

for all $x_{1:t-|x|} \in \text{tpsupp}_{t-|x|}(X)$.

9.2 The calculateProspectiveReserve-function

The following R-Code 9.1 shows the calculateProspectiveReserve-function that calculates $V_t\left(x\right)$ according to the formula of Definition 9.10.

```
calculateProspectiveReserve <- function(past, tau, payments,
                                        interests, parallel=F,
                                        track.progress=F) {
  #calculate t.e
  maturity <- length(tau)+1
  #check whether length(past) >= l.min
  for (time in 1:maturity) {
    if ((length(past)+time-1) < (payments[[time]]$height-1)) {
      str <- paste("past is too short to check the payments at",
                   time)
      stop(str)
    }
  }
  if (maturity > 1) {
    for (time in 1:(maturity-1)) {
      if ((length(past)+time-1) < (tau[[time]]$height-1)) {
        str <- paste("past is too short to cover the height of
                     the context tree at", time)
        stop(str)
      }
    }
  }
  #check whether length(interests) is correct
  #we need an interest rate for every future time point, i.e.
  #for t_s+1, ..., t_e
  if (length(interests) != length(tau)) {
    stop("length(interests) not equal to length(tau)")
  }
  #extract the alphabet from tau
  if (length(tau)>0) {
    alphabet <- names(tau[[1]]$transitions)
  #if t=t_e tau has length zero and the alphabet does not matter
  #in that case we set the alphabet to NULL
  } else {
    alphabet <- NULL
  }
  #initialize the prospective reserve
```

```
V <- 0
#we first deal with the case t=t_e separately
if (maturity==1) {
  return(getPayment(past=past, tree=payments[[1]]))
}
#if track.progress=T we measure and print computation times
if (track.progress) {
  print.at <- floor(quantile(x=1:length(alphabet)^(maturity-1),
                    probs=seq(from=0.01, to=1, by=0.01)))
  print(paste(length(alphabet)^(maturity-1),
              "iterations to calculate"))
  tic()
}
#we calculate each summand of the outer sum of the computation
#formula
#if parallel=F the computation is not parallelized
if (!parallel) {
  for (path.index in 1:(length(alphabet)^(maturity-1))) {
    #we combine the input past and the future to obtain the full
    #path
    path <- c(past, permutations(k=maturity-1, alphabet,
                                 replace=T, index=path.index))
    #initialize the inner sum of the computation formula
    sum <- 0
    #calculate the summands
    for (time in 1:maturity) {
      #calculate the discounting factor
      v <- calculateDiscountingFactor(interests=interests,
                                      time=time)
      #calculate the payment
      b <- getPayment(past=path[1:(time+length(past)-1)],
                      tree=payments[[time]])
      #add the summand to the inner sum
      sum <- sum+v*b
    }
    #initialize the product of the computation formula
    product <- 1
```

```
    #calculate the factors
    for (time in 1:(maturity-1)) {
      #calculate the transition probability
      prob <- getTransitionProbability(
        past=path[1:(time+length(past)-1)],
        future=path[time+length(past)], tree=tau[[time]])
      #multiply the factor by the product
      product <- product*prob
    }
    #add the summand to the outer sum
    V <- V+product*sum
    #measure and print computation time
    if (track.progress) {
      if (path.index %in% print.at) {
        print(paste(names(print.at)[path.index==print.at],
              "done"))
        timer <- toc()
        print(paste("time remaining:",
              round((100-which(path.index==print.at))
              *unname(timer$toc-timer$tic), digits=2), "sec"))
        tic()
      }
    }
}
#if parallel=T the iterations are evenly distributed over the
#user initiated CPU cluster
} else {
  V <- foreach(path.index=1:(length(alphabet)^(maturity-1)),
               .combine="+", .packages="arrangements",
               .export=c("calculateDiscountingFactor",
                         "getPayment",
                         "getTransitionProbability")) %dopar% {
    #we combine the input past and the future to obtain the full
    #path
    path <- c(past, permutations(k=maturity-1, alphabet,
                                 replace=T, index=path.index))
    #initialize the inner sum of the computation formula
```

```
      sum <- 0
      #calculate the summands
      for (time in 1:maturity) {
        #calculate the discounting factor
        v <- calculateDiscountingFactor(interests=interests,
                                        time=time)
        #calculate the payment
        b <- getPayment(past=path[1:(time+length(past)-1)],
                        tree=payments[[time]])
        #add the summand to the inner sum
        sum <- sum+v*b
      }
      #initialize the product of the computation formula
      product <- 1
      #calculate the factors
      for (time in 1:(maturity-1)) {
        #calculate the transition probability
        prob <- getTransitionProbability(
          past=path[1:(time+length(past)-1)],
          future=path[time+length(past)], tree=tau[[time]])
        #multiply the factor by the product
        product <- product*prob
      }
      #add the summand to the outer sum
      product*sum
    }
  }
  #return the prospective reserve
  return(V)
}
```

R-Code 9.1: The calculateProspectiveReserve-function

Via past the user has to input a state change sequence x as a vector of state names. Remember that x has to be of sufficient length, i.e. $|x|$ must be greater than or equal to $l^{\min}$.
tau has to be a list of length $(t_e - t_s)$ that contains the estimated context trees at times $t_s, ..., t_e - 1$ of the underlying model fit in the given order.

Using interests the user has to input a $(t_e - t_s)$-dimensional vector of the risk-free interest rates r_{t_s+1}, ... , r_{t_e} in the given order.
While parallel defaults to false, the user can switch to a parallelized computation by handing over parallel=T. Each summand of the outer sum of the computation formula given in Definition 9.10 can be calculated independently. Therefore their calculation can be distributed (and is distributed) evenly over the available CPU cores. In R the user can initiate the CPU-cluster using e.g. the doParallel-package, cf. Microsoft Corporation and Weston (2019), or the doMPI-package, cf. Weston (2013).
By setting track.progress to T the estimated remaining computation time and the progress (measured in percent) are printed to the console.
Recall the short notation of payments, cf. Definition 9.2. This notation implies a hierarchical structure, as it was the case for the context functions too. Thus it is convenient to specify the (non-zero) payments at t as a hierarchical tree.

Example 9.11. Assume a contract would unify all payments presented in Example 9.3. The non-zero payments at a time $t_s + 5 \leq t \leq t_e$ can be visualized as shown in the following Figure 9.1. The nodes of the tree specify the sequences triggering payments and the amount of the actual payment is specified via the payment-attribute of the node.

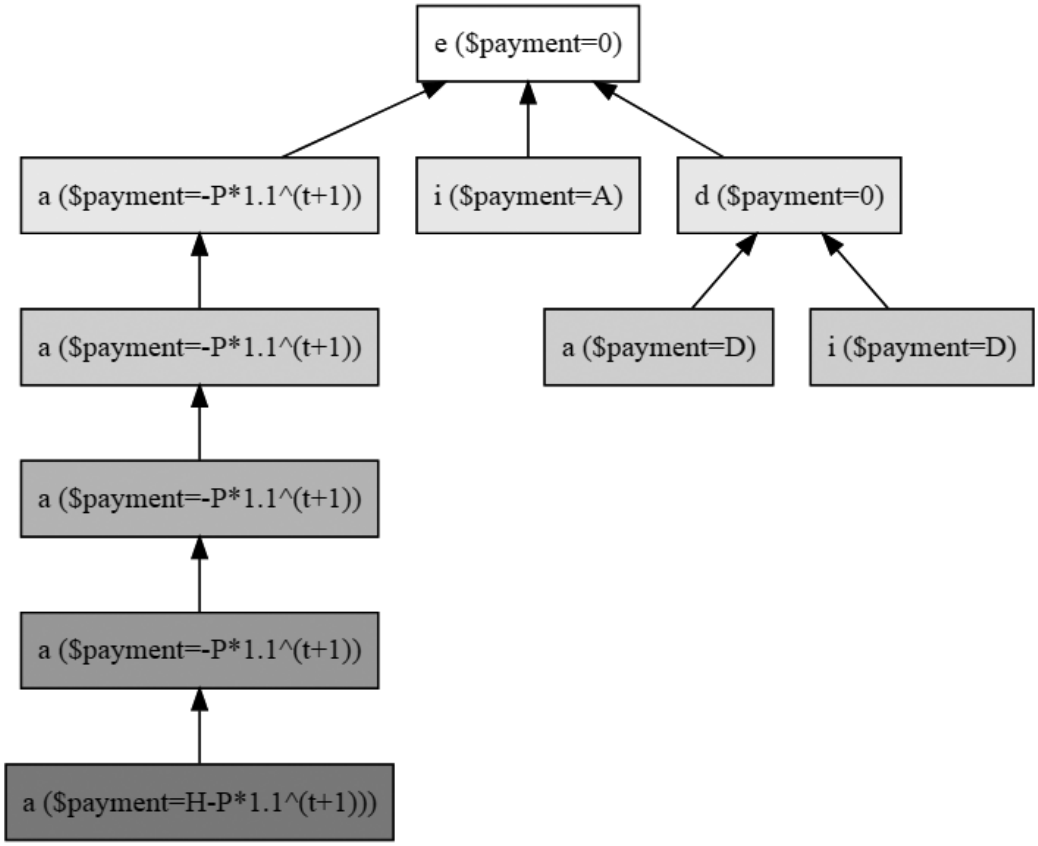

Figure 9.1: Visualizing the payments of Example 9.3 as a tree. In addition to the node itself its payment-attribute is printed in brackets.

In R the payment tree (here called tree) would be initialized as in the following R-Code 9.2. The data.tree-package is used, cf. Glur (2018).

```
#initialize the root node
tree <- Node$new("e")
tree$payment <- 0
#add the sequences and payments of (a)
tree$AddChild("d")
tree$"d"$payment <- 0
tree$"d"$AddChild("a")
tree$"d"$"a"$payment <- D
tree$"d"$AddChild("i")
tree$"d"$"i"$payment <- D
#add the sequences and payments of (b)
```

```
tree$AddChild("i")
tree$"i"$payment <- A
#add the sequences and payments of (c)
tree$AddChild("a")
tree$"a"$payment <- -P*1.1^(t+1)
tree$"a"$AddChild("a")
tree$"a"$"a"$payment <- -P*1.1^(t+1)
tree$"a"$"a"$AddChild("a")
tree$"a"$"a"$"a"$payment <- -P*1.1^(t+1)
tree$"a"$"a"$"a"$AddChild("a")
tree$"a"$"a"$"a"$"a"$payment <- -P*1.1^(t+1)
#add the sequences and payments of (d)
tree$"a"$"a"$"a"$"a"$AddChild("a")
tree$"a"$"a"$"a"$"a"$"a"$payment <- H-P*1.1^(t+1)
```

R-Code 9.2: Defining the payment tree at $t_s + 5 \leq t \leq t_e$ of the payments of Example 9.3 in R

Via the payments-input the user has to handover a list of length $(t_e - t_s + 1)$ containing the payment trees at times t_s, ..., t_e in the given order. Each payment tree has to be defined as shown in Example 9.11.
The calculateProspectiveReserve-function depends on the calculateDiscountingFactor- and getPayment-function, cf. R-Codes 9.3 and 9.4.
calculateDiscountingFactor uses the input interests to compute the discounting factors appearing in the computation formula of Definition 9.10. The relationship (9.2) between the discounting factor and the risk-free interest rate is used.

```
calculateDiscountingFactor <- function(interests, time) {
  if (time==1) return(1)
  #initialize the discounting factor
  v <- 1
  #calculate the one-step discounting factors
  for (t in 2:time) {
    interest <- interests[t-1]
    #calculate the multiple-step discounting factor
    v <- v*(1/(1+interest))
```

```
  }
  #return the discounting factor
  return(v)
}
```

R-Code 9.3: The calculateDiscountingFactor-function

```
getPayment <- function(past, tree) {
  #in the following we switch between child and parent nodes
  parent <- tree
  #initialize the payment
  payment <- NULL
  #we navigate down the branch of the tree given by past
  for (i in length(past):1) {
    state <- past[i]
    child <- parent[[state]]
    #if the whole past is not contained as a branch, we will
    #encounter a missing child
    if (is.null(child)) {
      #then the relevant past is the path from the root to the
      #parent of the missing child
      payment <- parent$payment
      break
    }
    parent <- child
  }
  #if we do not encounter a missing child, the whole past is
  #contained in a branch of the tree
  #in this case the relevant past is the whole past
  if (is.null(payment)) payment <- child$payment
  #return the payment
  return(payment)
}
```

R-Code 9.4: The getPayment-function

getPayment gets handed a payment tree and a state change sequence. It identifies the node corresponding to the input sequence and outputs the payment-attribute

of the identified node. Most of getPayment is completely analogous to getTransitionProbability, cf. R-Code 7.13, but instead of the transition probability we extract the payment.

9.3 The net premium

One of the reasons why the prospective reserve is such a key quantity in insurance mathematics is that it constitutes an elegant possibility of calculating the net premium π^{net} (recall Chapter 1.1): Insurance contracts usually only get sold to individuals that are and have been healthy, and it is highly uncommon that a complementary disability insurance is sold to an already disabled individual. Since the disabled would instantly be eligible for the disability annuity, such a business deal would most likely be economically disastrous for the insurance. While there might be a small chance that the disabled will recover and start paying premiums, the insurance is clearly better off only accepting healthy new customers (or excluding any pre-existing conditions from the coverage as it is usually done in practice). The premium is paid to the insurance company by the insurance holder, so it is part of one of the payment functions introduced in Definition 9.1. In most realistic cases the premium gets paid as long as the insurance holder is healthy and not eligible for the insurance benefit, therefore the premium is usually part of a sojourn payment. Now recall that the net premium π^{net} is the premium determined by the equivalence principle, as it was established in the introductory Chapter 1.1. Therefore one can calculate (and define) the net premium as follows:

Definition 9.12. Let $x \in M^{|x|}$ be a state sequence of length

$$t_s \geq |x| \geq l^{\min}$$

with $\mathbb{P}\left(X_{t_s-|x|+1:t_s} = x_{t_s-|x|+1:t_s}\right) > 0$ that is accepted by the insurance company to purchase the coverage. For a premium π_0 that satisfies

$$V_{t_s}(x) = 0 \tag{9.6}$$

we call $\pi^{\text{net}} := -\pi_0$ net premium.

The defining equation (9.6) ensures that the expected discounted future payments of the contract are zero at the beginning of the time period t_s in which the contract starts. This is exactly the equivalence principle: The expected costs for the

insurance and the premiums paid by the insured balance out to zero.
Although we do not indicate this in our notation, π^{net} is in fact allowed to depend on t. It is not required to be constant in t.

Example 9.13. In Chapter 8.1.1 we fitted and tuned tiVLMC-models to the data set data consisting of $n = 100{,}000$ observations of length $T = 4$ with values in $M = \{a, i, i_+, d\}$. For the BIC-tuned model (that corresponded to the cutoff $C_{\text{BIC}} = 1$), let us consider the following contract:

- The coverage is sold to active insurance holders at $t_s = 1$ and matures at $t_e = 4$.
- The risk-free interest rates are $r_2 = 2.00\%$, $r_3 = 1.00\%$ and $r_4 = 0.50\%$.
- As long as the insurance holder is active, she or he has to pay a premium. At $t = 1$ the premium is π^{init} monetary units and it increases at a rate of 10%, i.e.
$$b_t(a) = 1.1^{t-1}\pi^{\text{init}}$$
for $1 \leq t \leq 4$.
- Insurance holders transitioning from active to being invalid or heavily invalid receive a payment of two monetary units in order to cover the one-time expenses for e.g. house modifications. I.e.
$$b_t((i, a)) = b_t((i_+, a)) = 2$$
for $2 \leq t \leq 4$.
- Insurance holders that have already been impaired at the last time point receive a disability annuity of one monetary unit if they are invalid and one and a half monetary units if they are heavily invalid. I.e.
$$b_t((i, i)) = b_t((i, i_+)) = 1$$
and
$$b_t((i_+, i)) = b_t((i_+, i_+)) = 1.5$$
for $2 \leq t \leq 4$.

Running R-Code 9.5 we now calculate the initial premium π^{init} so that

$$-1.1^{t-1}\pi^{\text{init}}$$

is a net premium.

```
#calculate the estimated context tree
estimate <- estimateTau(data=data, measure="KLD", cutoff=1)
#the prospective reserve as a function of the initial premium
V.pi.init <- function(pi.init) {
  #define the payment trees
  payments <- list()
  for (t in 1:4) {
    #initialize the root node
    payments[[t]] <- Node$new("e")
    payments[[t]]$payment <- 0
    #premium
    payments[[t]]$AddChild("a")
    payments[[t]]$"a"$payment <- 1.1^(t-1)*pi.init
    if (t >= 2) {
      #disability annuity for i->i and i+->i
      payments[[t]]$AddChild("i")
      payments[[t]]$"i"$payment <- 0
      payments[[t]]$"i"$AddChild("i")
      payments[[t]]$"i"$"i"$payment <- 1
      payments[[t]]$"i"$AddChild("i+")
      payments[[t]]$"i"$"i+"$payment <- 1
      #disability annuity for i->i+ and i+->i+
      payments[[t]]$AddChild("i+")
      payments[[t]]$"i+"$payment <- 0
      payments[[t]]$"i+"$AddChild("i+")
      payments[[t]]$"i+"$"i+"$payment <- 1.5
      payments[[t]]$"i+"$AddChild("i")
      payments[[t]]$"i+"$"i"$payment <- 1.5
      #payment for a->i and a->i+
      payments[[t]]$"i"$AddChild("a")
      payments[[t]]$"i"$"a"$payment <- 2
```

```
      payments[[t]]$"i+"$AddChild("a")
      payments[[t]]$"i+"$"a"$payment <- 2
    }
  }
  #calculate the prospective reserve depending on the initial
  #premium
  V <- calculateProspectiveReserve(past="a", tau=estimate$tau,
                                   payments=payments,
                                   interests=c(0.02, 0.01, 0.005))
  return(V)
}
#calculating the zeros of above function on [-10,0],
#i.e. the initial premium that yields a prospective reserve of
#zero
pi.init <- uniroot(f=V.pi.init, interval=c(-10, 0))$root
```

R-Code 9.5: Calculating the prospective reserve $V_1(a)$ and the net premium π^{net}

To do so, we use the native R-function uniroot that here outputs the roots of $V_1(a)$ viewed as a function of π^{init}. We limit the numerical root search to the interval $[-10, 0]$.
We obtain $\pi^{\text{init}} = \text{pi.init} = -0.56$ (rounded to two decimal digits). Therefore the net premium is given by

$$\pi^{\text{net}} = -1.1^{t-1} \cdot (-0.56) = \begin{cases} 0.56, & t = 1 \\ 0.62, & t = 2 \\ 0.68, & t = 3 \\ 0.75, & t = 4. \end{cases}$$

10 The German long-term care insurance

In the following we are going to apply the developed tiVLMC fitting techniques to real life data. First we are going to briefly describe the data set and prepare it for the estimation of tiVLMC-models. After that we are going to perform the actual model fitting. We are going to end this chapter by calculating net premiums and some prospective reserves for an insurance contract using the tiVLMC-models.

10.1 A brief data description

The data set is provided in an anonymized form by the German insurance corporation SIGNAL IDUNA, cf. SIGNAL IDUNA (2020). It contains the documentation of the nursing care level of 684,384 private disability insurance holders on a monthly basis within the time frame from January 2005 to July 2014.
Under German law a person is called disabled if impairments of his or her self-reliance or abilities in general caused by health issues make the help of others necessary. It is also required that the person is not able to countervail the physical, cognitive or psychological impairment on his or her own. Lastly, the impairment must be of a non-temporary nature and its severity has to exceed a defined lower bound. This is regulated by § 14 SGB XI, cf. Bundesamt für Justiz (2020). The severity of the disability is categorized into several levels. The level determines the amount of the payment the disabled insurance holder receives from the insurance company. At the start of the observation there were four levels of disability named “Pflegestufe 1”, “Pflegestufe 2”, “Pflegestufe 3” and “Pflegestufe 3 mit Härtefall”, where the degree of disability increases from Pflegestufe 1 (lowest) to Pflegestufe 3 mit Härtefall (highest). During the observation period the “Pflegestufe 0” was introduced in 2013 as a part of the “Gesetz zur Neuausrichtung der Pflegeversicherung” reform, cf. Bundestag (2012). The Pflegestufe 0 was positioned in-between not being disabled and the Pflegestufe 1. It was intended to offer financial support for dementia patients that did not qualify for Pflegestufe 1 or higher. Then this categorization was completely revised and replaced in 2017 via the “Zweites Pflegestärkungsgesetz” reform, cf. Bundestag (2015). Since this replacement the severity of disability is categorized in so-called “Pflegegrade”, cf. Bundesamt für Justiz (2017). Thus the state space of the data set, settled in the Pflegestufen-setting, consists of seven states: active (i.e. not disabled), Pflegestufe 0, Pflegestufe 1, Pflegestufe 2, Pflegestufe 3, Pflegestufe 3 mit Härtefall and deceased.

Furthermore the data is subject to censoring: Not all individuals under study are insured over the complete observation period, some terminate the contract early or exit the study for other, unknown reasons.
Additionally to their health status, an identification number, their gender, their year of birth and the year the insurance contract began are recorded for each insurance holder.
Since our main focus lies on demonstrating the developed tiVLMC-techniques, we here limit ourselves to this short description. However, we refer the reader to chapter 3 of Hess (2015) for a more detailed description of the data set. Analyzing the data set Hess (2015, pages 17-19) was able to uncover four different inconsistencies. After consultation with the SIGNAL IDUNA Hess (2015) was able to remove each one of them and re-establish the consistency of the data. As part of this data processing the last three months of the observation period, May 2014, June 2014 and July 2014, had to be removed from the data set, thus the observation period of the processed data consists of 112 months (January 2005 to April 2014). Our further data processing steps are based on this pre-processed data of Hess (2015).

10.2 Data preparation

Our modelling goal is to predict the future health status X_{t+1} of an insured of age $t+1$ based on his health development $(X_1, ..., X_t)$ up to his present age t. To achieve this goal we are going to train a tiVLMC-model X on the data. We are looking for the health status sequences that are of high significance when it comes to predicting the future health development. I.e. we are interested in the contexts of this tiVLMC.
The birth years of the insurance holders vary from 1899 to 2014 whereas the observation period is the same for every insurance holder. Thus one data column contains the health statuses of insurance holders of different ages. To function as training data for our model, it is therefore necessary to rearrange the data so that all insurance holders are of the same age in each column. The major problem we are facing is that the health statuses are recorded monthly, while only the year of birth and not the birth month is known. Hence a straightforward shifting of the data is not possible and assumptions concerning the birth month of every individual have to be made. We choose to enrich the data by assuming that all insurance holders are born in July. We count the age in months starting at one (and not at zero), e.g. we assume that an insured born in 1950 is of age one in

July 1950, of age two in August 1950 etc. By making this assumption we are able to shift the observations, so that every column contains insurance holders of the same age.

This time shift from calendar time to age is inevitable to achieve our modelling goal. Without this change the data is not of the correct format to serve as training data. However the time change also implies two new inconsistencies:

First, the intensity at which the observations within one data column are affected by calendar time trends now differs. The data is already subject to calendar time trends before the shift, but there each column is affected at the same intensity. We reduce this effect by limiting ourself to the more homogeneous sub data set of 38,236 insured born from 1940 to 1944. A person born in July 1940 is 775 months (64 years) old at the start of the observation in January 2005 and 886 months (73 years) old at the end of the observation in April 2014. A person born in July 1944 is 727 months (60 years) old at the start of the observation in January 2005 and 838 months (69 years) old at the end of the observation in April 2014. Thus our data now documents the monthly health statuses of insured aged from 727 to 886 months and it is of dimension $38{,}236 \times 160$. It is reasonable to assume for this five-year cohort that the extent to which observations are affected by calendar time trends is similar: The life expectancy, the available medical treatment methods etc. should be similar.

However, the political environment is not the same for all observations of one data column and this is the second inconsistency. Since the Pflegestufe 0 was introduced during the observation, it is possible that an insured aged t is assigned Pflegestufe 0 while another earlier born insured aged t with the exact same health condition was classified as healthy since the Pflegestufe 0 was not yet introduced at the calendar time that the second insured hit age t. We eliminate this inconsistency by removing the Pflegestufe 0 and replacing its occurrences with active. I.e. we standardise the categorization of disability to the laws in force at the start of the observation in January 2004. We want to note that we expect the information lost due to this processing step to be minimal: The Pflegestufe 0 only occurred at 629 out of the total $38{,}236 \cdot 160 = 6{,}117{,}760$ data points.

Restricting ourselves to the sub data set obviously also reduces the computational effort needed to perform our model fitting and follow-up calculations.

The restriction is practicable: One can fit a tiVLMC for every five-year cohort. Then, when dealing with a new customer, we use the model that corresponds to her or his birth year to make the predictions about her or his future health development. Here we have exemplarily chosen the birth years 1940 to 1944

because the insured born in this time frame are of an interesting age according to changes in health at the time of the observation. They are neither in the high ages where getting disabled becomes an almost sure event nor are they so young that getting disabled is completely rare. For reference, in 2017 six percent of the 70- to 74-year-old Germans were eligible for disability benefits, cf. Statistisches Bundesamt (2018).
To summarize, this data preparation yields a data set data.signal of 38,236 observations of length 160. Each observation documents the 160 monthly health statuses of an insured at the ages 727 to 886. Possible values for the health status are active, Pflegestufe 1, Pflegestufe 2, Pflegestufe 3, Pflegestufe 3 mit Härtefall and deceased. As usual censored data points are assigned NA. Using data.signal as training data, the state space of our model thus consists of six states. We abbreviate them as displayed in the following Table 10.1.

state	abbreviation
active	a
Pflegestufe 1	1
Pflegestufe 2	2
Pflegestufe 3	3
Pflegestufe 3 mit Härtefall	3h
deceased	d

Table 10.1: The abbreviations of the state names in data.signal

Note that we have not split the data into male and female insurance holders. Pricing insurance contracts differently depending on gender is forbidden in Germany by the "Equal Treatment in Goods and Services Directive 2004" EU directive, cf. European Court of Justice (2004).
The following R-Code 10.1 shows the most frequently observed health path. Since most of the insured in data.signal are born in 1941 (7,740 in 1940, 8,033 in 1941, 6,908 in 1942, 7,744 in 1943 and 7,811 in 1944) and since we expect being active to be more likely than being disabled, it is no big surprise that this health path is an insured born in 1941 that is active during all of the 112 months of the observation period, where he is aged 763 to 874. 4,331 insurance holders move through this exact path.

```
data.signal[1627, ]

## 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742
##  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758
##  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774
##  NA  NA  NA  NA "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a"
## 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790
## "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a"
## 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806
## "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a"
## 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822
## "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a"
## 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838
## "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a"
## 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854
## "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a"
## 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870
## "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a"
## 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886
## "a" "a" "a" "a"  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
```

R-Code 10.1: The most frequent observation in the data.signal data set

10.3 The tiVLMC-models

Now we tune and fit the tiVLMC-model using the tuneC-function explained in Chapter 7.4 and R-Code 7.11 and exemplarily applied in R-Code 8.12. We compute AIC and BIC on the lattice

$$\mathscr{C} = \{0.001, 0.002, ..., 0.999, 1.000\}$$

of 1,000 equidistant knots. The input options we choose here are data=data.signal, cutoffs=$\mathscr{C}$, measure="KLD" and parallel=T.
The fitting was performed using 500 CPU cores of the CARL High-Performance Computing cluster. The cluster nodes we requested consist of Intel Xeons (E5-2650 v4), each CPU possessing 12 cores clocked at 2.2GHz, cf. Harfst (2020).

The computation took 4 hours and 27 minutes to be completed. Since tuneC first considers the smallest cutoff value and then distributes the other cutoffs evenly over the available CPU cores, one can estimate the computation of a single model fit to require about

$$\frac{4 \cdot 60\,\text{minutes} + 27\,\text{minutes}}{3} = 89\,\text{minutes}.$$

The result of the tuning process is graphically displayed in the following Figure 10.1. The continuous plotted blue line is the information criterion, i.e. AIC or BIC respectively. The vertical grey bars indicate the sampling rate: Within each segment the information criterion was computed for 50 equidistant values of C. The vertical red bar indicates the cutoffs C_{AIC} or C_{BIC} that minimize AIC or BIC respectively, cf. (6.4), on $\mathscr{C}$.

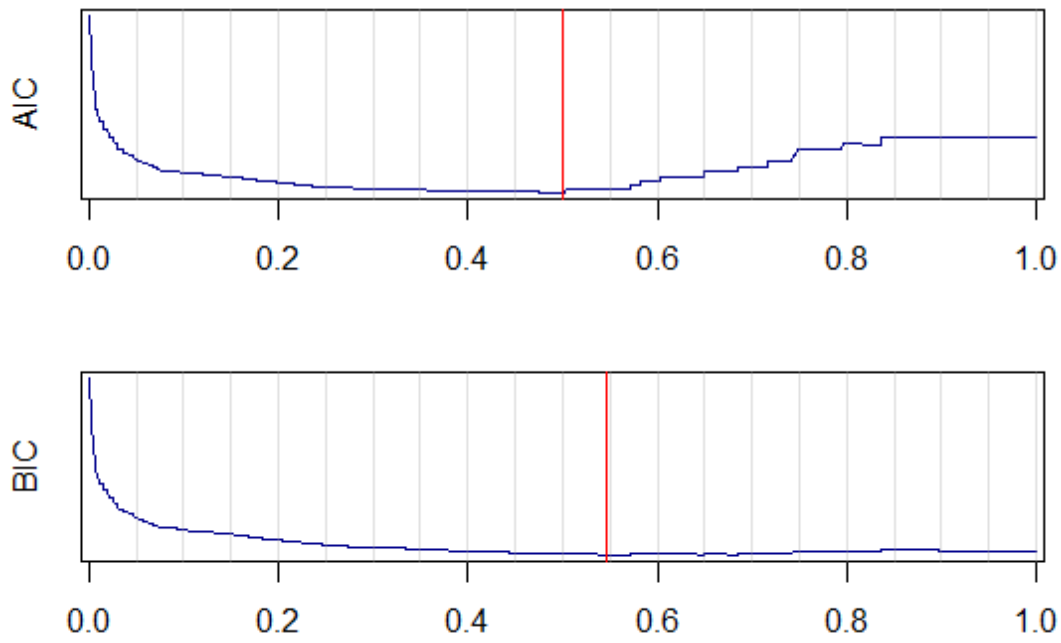

Figure 10.1: Tuning Algorithm 5.6 on the data.signal data set with AIC and BIC

In numbers, the tuning process yields the results

$$C_{\text{AIC}} = 0.501,$$
$$C_{\text{BIC}} = 0.547$$

and we will now take a closer look at the AIC- and BIC-tuned models.
Figure 10.2 plots the tree heights, i.e. the order $|\tilde{c}_t|$ of the estimated context

function $\tilde{c}_t$ at $t \in \{727, ..., 885\}$. In both models this maximal tree height varies between one and two. Hence the order of both tiVLMC-models is two and they could both be embedded into full second-order time-discrete Markov processes. The tree heights of the AIC-tuned model are 104 times two and 55 times one. In the BIC-tuned model 102 trees are of height two and the remaining 57 are of height one. At this point one can already anticipate that a first-order time-discrete Markov process might be an inappropriate model: Both estimated context trees are of height two at the majority of time.
At every time point where the tree height of the BIC-tuned model is two, this is also the case in the AIC-tuned model. As in the example in Chapter 8.2.1, the BIC-tuned model is less complex than the AIC-tuned model. This is backed up by theory: A sequence of increasing cutoffs leads to nested models that decrease in complexity.

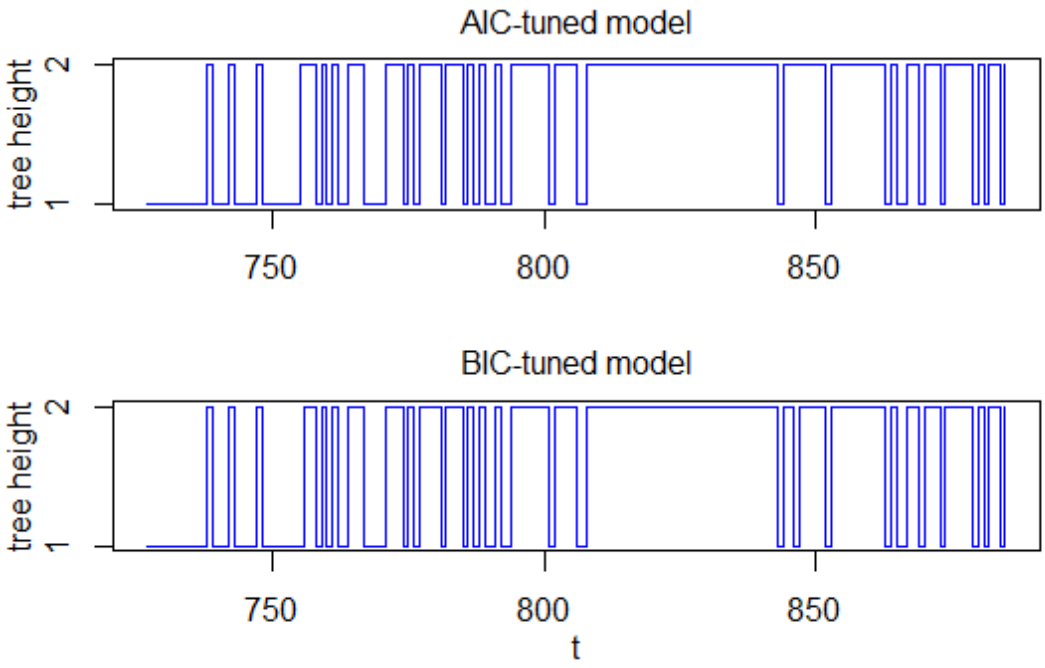

Figure 10.2: The tree heights $|\tilde{c}_t|$ of the estimated context trees $\tilde{\tau}_t$ at t of the AIC- and the BIC-tuned tiVLMC-model

The AIC-tuned model's complexity amounts to 5,580 compared to the lower value of 5,535 for the BIC-tuned model. For comparison: The embedding full second-order time-discrete Markov process would have 28,475 model parameters, cf. (3.2). The number of nodes within each estimated context tree is higher in the AIC-tuned model than in the BIC-tuned model, cf. Figure 10.3. The log-likelihood of the AIC-tuned model is -60,344.41, the BIC-tuned model's is -60,468.86 (numbers

rounded to two decimal digits). This yields the minimum of 173,030.27 of BIC and 131,848.82 of AIC in the corresponding models.

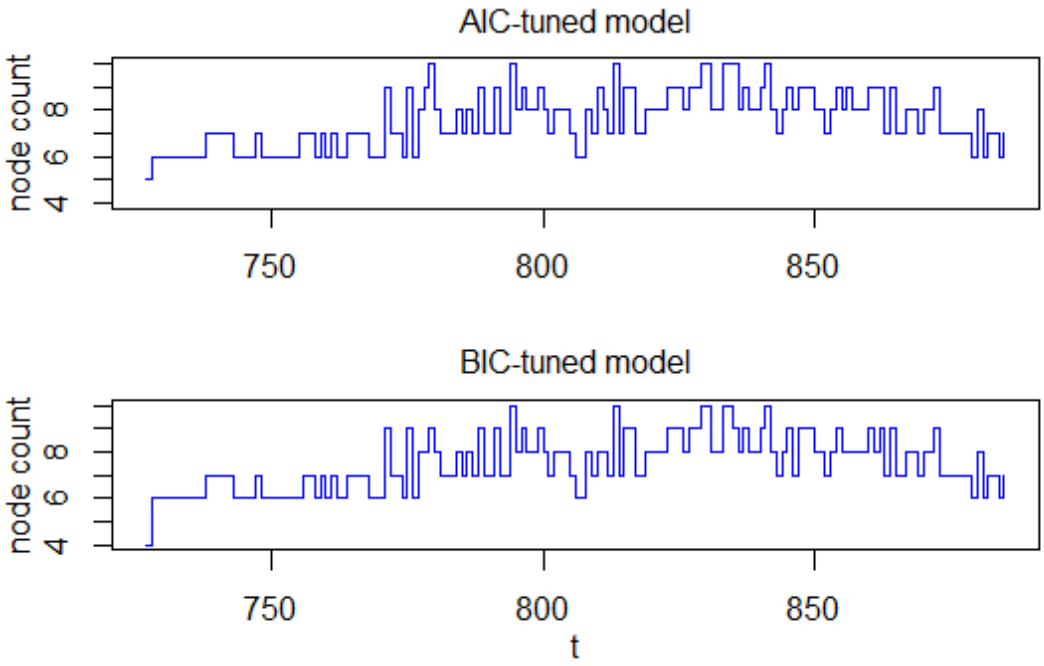

Figure 10.3: The node counts of the estimated context trees $\tilde{\tau}_t$ at t of the AIC- and the BIC-tuned tiVLMC-model

Now we visualize the fitted models. In the AIC-tuned fit 41 different trees appear as context tree estimates. In the BIC-tuned fit 39 different trees are estimated. Both models have 37 trees in common, hence the total number of occurring trees amounts to 43. We label these 43 trees with the numbers from one to 43. The following Figure 10.4 displays the movement of each model from context tree estimate to context tree estimate as the age t increases.

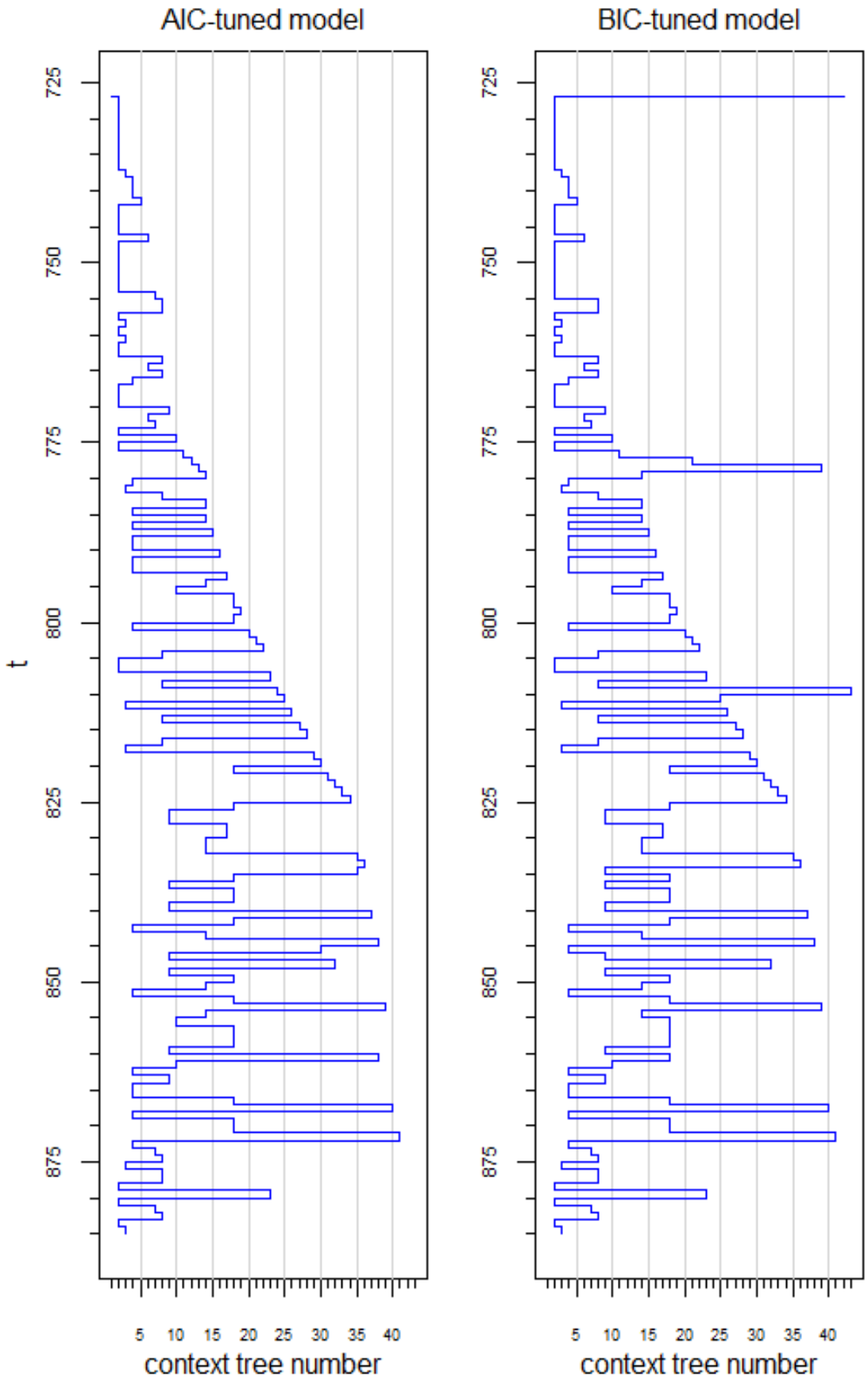

Figure 10.4: The movement of the estimated context tree $\tilde{\tau}$ between the 43 occurring trees as the time t passes

The visualization of the fitted models is completed by listing all 43 occurring context tree estimates in the Figures B.1-B.43. Since this list takes up a lot of space, we postpone printing it to the Appendix B. Via a three dimensional visualization the context tree movement can be illustrated in a much clearer way: At

https://cloud.uol.de/s/yZCGEQzriK5bJDD

we provide a short mp4-video clip that shows the development of the context tree of the AIC-tuned model as the age t increases in the cloud storage of the University of Oldenburg. Accessing this file requires the password MzxEbjnrKfi4.
If we fit the model using the L^1-norm, cf. Chapter 5.3, i.e. if we switch to using measure="L1" instead of measure="KLD" while keeping the other input arguments and the cutoff-lattice unchanged, the tuning process yields

$$C_{\text{AIC}} = 0.033,$$
$$C_{\text{BIC}} = 0.346,$$

cf. Figure 10.5. We kept the computational environment unchanged. The computation time was 13 minutes longer (4 hours and 40 minutes) compared to the time needed in the Kullback-Leibler case.
The tree heights $|\tilde{c}_t|$ are again plotted in Figure 10.6 and the node counts in Figure 10.7.

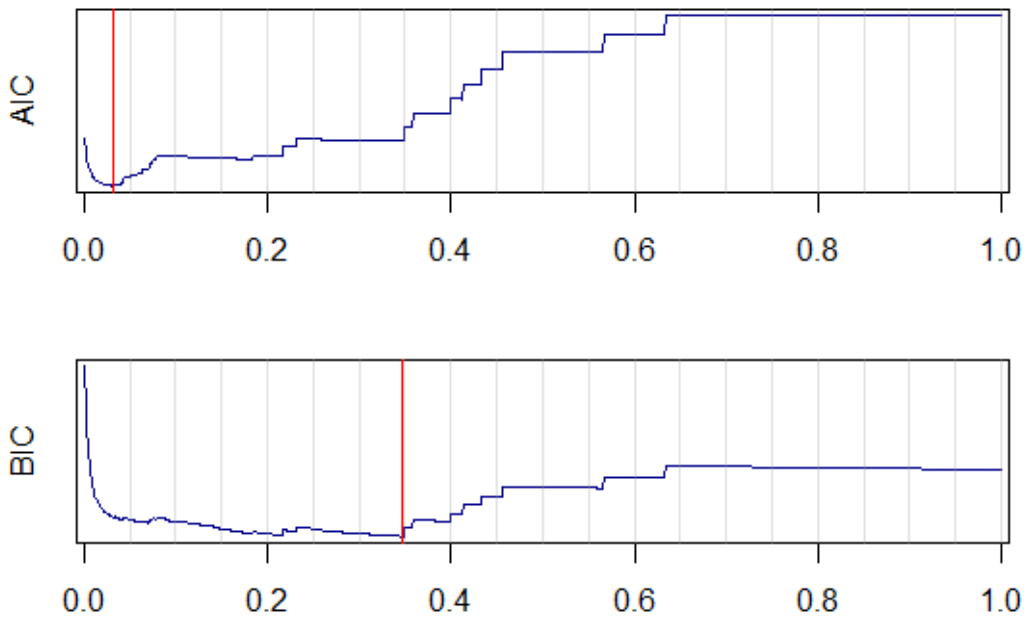

Figure 10.5: Tuning Algorithm 5.6 on the data.signal data set with AIC and BIC and using the L^1-norm instead of the Kullback-Leibler divergence

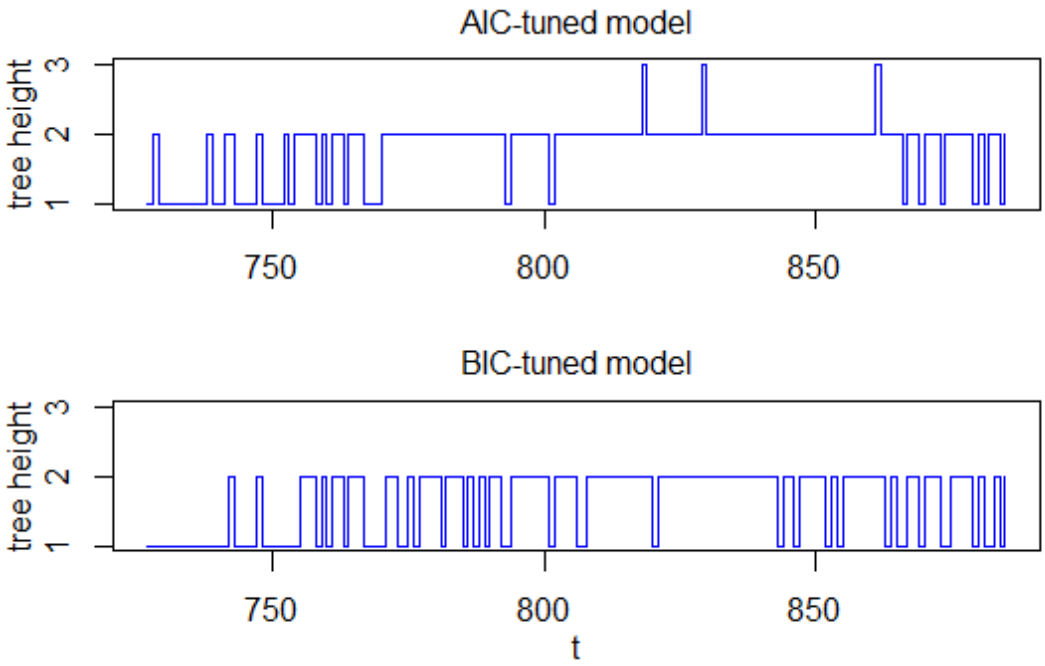

Figure 10.6: The tree heights $|\tilde{c}_t|$ of the estimated context trees $\tilde{\tau}_t$ at t of the AIC- and the BIC-tuned tiVLMC-model calculated using the L^1-norm

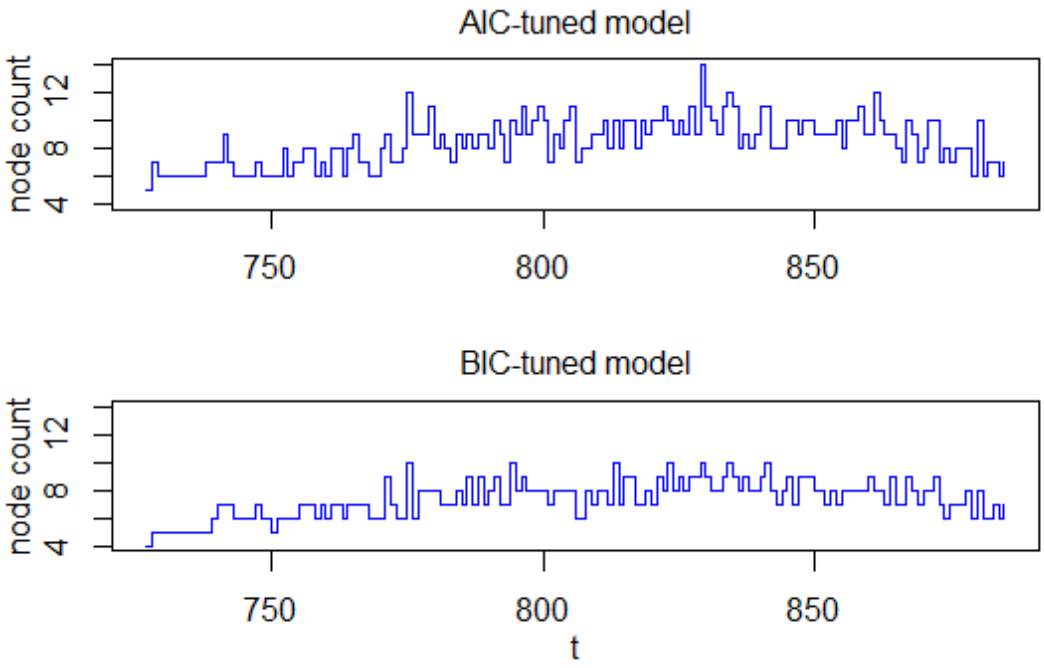

Figure 10.7: The node counts of the estimated context trees $\tilde{\tau}_t$ at t of the AIC- and the BIC-tuned tiVLMC-model calculated using the L^1-norm

As in the measure="KLD" scenario, the tree heights of the BIC-tuned model have two as their upper limit. This means that the BIC-tuned tiVLMC-model as well as its embedding full time-discrete Markov process are of order two. However, the tree heights of the AIC-tuned model hit three at three different ages, namely at 818, 829 and 861. While the context tree at those times takes three different forms, the sequence responsible for the tree height exceeding two is $(1, a, 1)$ at all three ages. The order of the BIC-tuned tiVLMC is three.

The AIC-tuned model has a model complexity of 6,270 and a log-likelihood of -60,059.17. The BIC-tuned model has a model complexity of 5,460 and a log-likelihood of -62,689.80. The embedding full third-order time-discrete Markov process of the AIC-tuned tiVLMC would have 169,775 model parameters, cf. (3.2). The minimal AIC is 132,658.34 and the minimal BIC 176,766.31 respectively.

As one already might expect by looking at the higher model complexity, the number of different context tree shapes is higher than when we used the Kullback-Leibler divergence: Now a total of 93 different trees occurs, more than double the amount in the Kullback-Leibler setting.

We summarize the characteristics of the four fitted tiVLMC-models in the fol-

lowing Table 10.2.

	measure="KLD"		measure="L1"	
	AIC	BIC	AIC	BIC
model complexity	5,580	5,535	6,270	5,460
number of different trees	41	39	77	45
order of the tiVLMC	2	2	3	2
model complexity of the embedding	28,475	28,475	169,775	28,475
complexity reduction	80.40%	80.56%	96.31%	80.83%
number of different contexts	16	16	16	16

Table 10.2: An overview of the tiVLMC-model fitting

In the fifth row of Table 10.2 we calculate the percentage of less model parameters of the tiVLMC-model compared to its embedding time-discrete full Markov process. By using a tiVLMC as our model of choice instead of a classical time-discrete Markov process of sufficient order the number of model parameters decreases by at least 80%. Despite being less complex, Corollary 5.9 states that the tiVLMC-model asymptotically still catches all dependencies in the data as the number of observations diverges.
In the last row we print the total number of different contexts

$$\left|\left\{\tilde{c}_t(w) : w \in M^t,\, 727 \leq t \leq 885\right\}\right|$$

that appear in the tiVLMC-model. As a coincidence this number is 16 in all four cases. Thus, when one tries to predict the future health status X_{t+1} of an insured at age t, it is sufficient to consider 16 different pasts and not all possible pasts of length t are needed. We list these 16 contexts in Table 10.3. Remember that we use a reversed notation, i.e. the most recent state of a state change sequence is positioned on the left.

	contexts of length l			
	measure="KLD"		measure="L1"	
l	AIC	BIC	AIC	BIC
1	(a)	(a)	(a)	(a)
	(1)	(1)	(1)	(1)
	(2)	(2)	(2)	(2)
	(3)	(3)	(3)	(3)
	$(3h)$	$(3h)$	$(3h)$	$(3h)$
	(d)	(d)	(d)	(d)
2	$(a,1)$	$(a,1)$	$(a,1)$	$(a,1)$
	$(a,2)$	$(a,2)$	$(a,2)$	$(a,2)$
	$(a,3)$	$(a,3)$	$(a,3)$	$(a,3)$
	$(1,a)$	$(1,a)$	$(1,a)$	$(1,a)$
	$(1,2)$	$(1,2)$	$(1,2)$	$(1,2)$
	$(2,a)$	$(2,a)$	$(2,a)$	$(2,a)$
	$(2,1)$	$(2,1)$	$(2,1)$	$(2,1)$
	$(3,a)$	$(3,a)$	$(3,a)$	$(3,a)$
	$(3,1)$	$(3,1)$	$(3,2)$	$(3,1)$
	$(3,2)$	$(3,2)$		$(3,2)$
3			$(1,a,1)$	

Table 10.3: The contexts $c_t(w)$ of the fitted models, $w \in M^t$, $727 \leq t \leq 885$

By looking at the contexts we can see e.g. that a, 1, 2 and 3 are subject to duration effects. For better understanding let us consider an insurance holder who is in a at age t, i.e. $X_t = a$. Then the following is true for all four tiVLMC-models: To predict her or his next health status X_{t+1}, it is relevant whether she or he was in Pflegestufe 1, Pflegestufe 2 or Pflegestufe 3 at $t-1$, i.e. we have to differentiate whether $X_{t-1:t} = (a,1)$, $X_{t-1:t} = (a,2)$ or $X_{t-1:t} = (a,3)$. If she or he was not in Pflegestufe 1, Pflegestufe 2 or Pflegestufe 3 at $t-1$, we do not have to differentiate between the three left over cases $X_{t-1:t} = (a,a)$, $X_{t-1:t} = (a,3h)$ and $X_{t-1:t} = (a,d)$. Rather we only have to consider that $X_t = a$. This can be justified as follows: A transition from d to a (resurrection) cannot occur. A recovery from $3h$ back to a is so unlikely that it is not considered as a context. Therefore out of the three cases, $X_{t-1:t} = (a,a)$ is so dominant that we need no further differentiation. The actual data does indeed confirm this: There is a total

of $38{,}236 \cdot (160-1) = 6{,}079{,}524$ transitions recorded in data.signal. 4,116,334 of them are uncensored (67.71%). No transitions from d into a are observed. Exactly one single transition from $3h$ to a can be found in the data compared to 3,878,641 transitions from a to a (94.23% of the uncensored transitions). Thus $(a, 3h)$ is not even relevant enough to be considered for pruning if nmin=2 is used when running Algorithm 5.6.

10.4 A comparison

As we have seen in the preceding Chapter 10.3, we obtain models that are of a much lower model complexity and simultaneously reflect all data dependencies by choosing tiVLMC as a replacement of classical higher-order time-discrete Markov processes. Within this Chapter we are going to evaluate how the prospective reserve, cf. Chapter 9, in a tiVLMC-model numerically differs from the prospective reserve in classical time-discrete Markov processes of different orders.

10.4.1 The contract

In our comparison we consider the following contract:

- The target group of the coverage are insurance holders that are

  $$t_s = 792$$

  months (66 years) old.

- To be eligible for the purchase, the potential buyer is required to be healthy at contract closure and in the five previous months.

- The coverage runs for 12 months and thus matures at

  $$t_e = 803.$$

- The risk-free interest rate is a constant two percent, i.e.

  $$r_t = 2\%$$

  for $t = 792, ..., 803$.

- At contract closure the insurance holder pays a one-time premium π, i.e.

$$b_{792}(a) = \pi.$$

- If an insured is disabled, a disability benefit will be paid out. The amount of the benefit depends on the level of disability:

$$\begin{aligned} b_t(1) &= 468, \\ b_t(2) &= 1{,}144, \\ b_t(3) &= 1{,}612, \\ b_t(3h) &= 1{,}995 \end{aligned}$$

for $t = 793, ..., 803$.

The benefits paid if an insured is disabled are denoted in Euro. They are in accordance with the real regulations in force in January 2005.
The length of the required health history $(a, a, a, a, a, a) = a^6$ is chosen so that it is as small as possible but still works with all models that are part of the comparison. We get back to this after explicitly proposing the candidate models in the following section.
This contract is convenient for us in two ways:
Firstly, it is a short-term contract. We choose a small duration period since it speeds up our numerical calculations while it does not impair the lesson to be learned in this chapter. We comment on the computation of long-term contracts in the conclusion, cf. Chapter 13.
Secondly, the premium π is a one-time payment at contract start. Mathematically this allows us to separate π in the calculation formula as follows:

$$\begin{aligned} & V_{792}\left(a^6\right) \\ =& \sum_{x_{793:803} \in M^{11}} \prod_{l=792}^{802} P_l\left(x_{l+1} \,\middle|\, c_l\left(x_{793:l} \cdot a^6\right)\right) \sum_{k=792}^{803} v_{792,\,k}\, b_k\left(x_{793:k} \cdot a^6\right) \\ =& \sum_{x_{793:803} \in M^{11}} \prod_{l=792}^{802} P_l\left(x_{l+1} \,\middle|\, c_l\left(x_{793:l} \cdot a^6\right)\right) \left(\pi + \sum_{k=793}^{803} v_{792,\,k}\, b_k\left(x_{793:k} \cdot a^6\right)\right) \\ =& \pi + \sum_{x_{793:803} \in M^{11}} \prod_{l=792}^{802} P_l\left(x_{l+1} \,\middle|\, c_l\left(x_{793:l} \cdot a^6\right)\right) \sum_{k=793}^{803} v_{792,\,k}\, b_k\left(x_{793:k} \cdot a^6\right). \end{aligned} \tag{10.1}$$

This separation enables us to obtain the net premium π^{net} by evaluating $V_{792}(a^6)$ (viewed as a function of π) only one single time: Calculate $V_{792}(a^6)$ with $\pi := 0$ and let V_0 denote the result. Then the net premium is given by

$$\pi^{\text{net}} = V_0. \tag{10.2}$$

For different premium schemes we might have to solve $V_{792}(a^6) = 0$ for π numerically, e.g. by using the uniroot-function as we did in Example 9.13. Calculating the roots of $V_{792}(a^6)$ numerically in general requires evaluating $V_{792}(a^6)$ for many different values of π. Thus going with the one-time premium the computation time can be reduced substantially.

10.4.2 The models

We consider seven different models trained on data.signal.
The first three models are tiVLMC-fits calculated in Chapter 10.3: the AIC-tuned tiVLMC using the Kullback-Leibler divergence for the tree pruning (KLD/AIC), the AIC-tuned tiVLMC (L1/AIC) and the BIC-tuned tiVLMC (L1/BIC) using the L^1-norm for the tree pruning. We do not consider the BIC-tuned tiVLMC using the Kullback-Leibler divergence for the tree pruning: it is identical to KLD/AIC during the contract duration.
Additionally we consider their embeddings, i.e. time-discrete Markov processes of orders two (M2) and three (M3).
For reference we include the industry standard into the comparison: a time-discrete Markov process of first order (M1). This is the most simple model.
Lastly, to cover the other end of the spectrum, we consider a time-discrete Markov process of sixth order (M6).
We require a potential buyer to be healthy at contract closure and in the five previous months, i.e. we require past $= (a, a, a, a, a, a) = a^6$. As one can see in Table 10.4 this is the minimal length that guarantees $|\text{past}| \geq l^{\min}$ at the age $t_s = 792$ at contract closure for all seven different models, where MC6 is the decisive factor.

		t											
		792	793	794	795	796	797	798	799	800	801	802	803
KLD/AIC	$\lvert c_t \rvert$	1	1	2	2	2	2	2	2	2	1	2	
	$\lvert b_t \rvert$	1	1	1	1	1	1	1	1	1	1	1	1
	$l^{\min}$	1											
L1/AIC	$\lvert c_t \rvert$	2	1	2	2	2	2	2	2	2	1	2	
	$\lvert b_t \rvert$	1	1	1	1	1	1	1	1	1	1	1	1
	$l^{\min}$	2											
L1/BIC	$\lvert c_t \rvert$	1	1	2	2	2	2	2	2	2	1	2	
	$\lvert b_t \rvert$	1	1	1	1	1	1	1	1	1	1	1	1
	$l^{\min}$	1											
MC1	$\lvert c_t \rvert$	1	1	1	1	1	1	1	1	1	1	1	
	$\lvert b_t \rvert$	1	1	1	1	1	1	1	1	1	1	1	1
	$l^{\min}$	1											
MC2	$\lvert c_t \rvert$	2	2	2	2	2	2	2	2	2	2	2	
	$\lvert b_t \rvert$	1	1	1	1	1	1	1	1	1	1	1	1
	$l^{\min}$	2											
MC3	$\lvert c_t \rvert$	3	3	3	3	3	3	3	3	3	3	3	
	$\lvert b_t \rvert$	1	1	1	1	1	1	1	1	1	1	1	1
	$l^{\min}$	3											
MC6	$\lvert c_t \rvert$	6	6	6	6	6	6	6	6	6	6	6	
	$\lvert b_t \rvert$	1	1	1	1	1	1	1	1	1	1	1	1
	$l^{\min}$	6											

Table 10.4: The required minimal length $l^{\min}$ of the health history past at contract closure at $t_s = 792$ for the different models

Note that e.g. for the KLD/AIC-model

$$V_{792}\left(a^6\right) = V_{792}(a)$$

holds, since $l^{\min} = 1$ at this age in this model. Thus including the most stringent requirement in the contract ensures the comparability between the different models and at the same time only affects each model to the necessary degree.

10.4.3 Net premiums

For each model we now calculate the net premium π^{net} of the contract. As we have already explained we obtain π^{net} by calculating V_0, recall (10.1) and (10.2). The following R-Code 10.2 exemplarily shows how π^{net} is calculated for the KLD/AIC-model. The calculation procedure is exactly the same for all seven models, the only difference is the context tree estimate handed over to the calculateProspectiveReserve-function as tau.

```
#age in months at the start of the contract
t.s <- 792
#contract duration
duration <- 12
#age in months at the end of the contract
t.e <- t.s+duration-1
#the contract runs from t.s, ..., t.e
#the interest rates at t.s+1, ..., t.e
interests <- rep(0.02, times=duration-1)
#we sell the contract to a person who was active at
#t.s-5, ..., t.s
past <- rep("a", times=6)
#the context trees at t.s, ..., t.e-1
tau <- estimate$tau[(t.s-727+1):(t.e-727)]
#the payment trees at t.s, ..., t.e as a function of the premium
paymentTree <- function(premium) {
  payments <- list()
  for (t in 1:duration) {
    #initialize the root node
    payments[[t]] <- Node$new("e")
    payments[[t]]$payment <- 0
    #the one-time premium has to be paid at the start of
    #the contract
    if (t==1) {
      payments[[t]]$AddChild("a")
      payments[[t]]$"a"$payment <- premium
    }
    #disability annuity in 1
```

```
    payments[[t]]$AddChild("1")
    payments[[t]]$"1"$payment <- 468
    #disability annuity in 2
    payments[[t]]$AddChild("2")
    payments[[t]]$"2"$payment <- 1144
    #disability annuity in 3
    payments[[t]]$AddChild("3")
    payments[[t]]$"3"$payment <- 1612
    #disability annuity in 3h
    payments[[t]]$AddChild("3h")
    payments[[t]]$"3h"$payment <- 1995
  }
  return(payments)
}
#the prospective reserve as a function of the premium
prospectiveReserve <- function(premium) {
  payments <- paymentTree(premium)
  V <- calculateProspectiveReserve(past=past, tau=tau,
                                   payments=payments,
                                   interests=interests)
  return(V)
}
#the prospective reserve for premium=0
V.0 <- prospectiveReserve(0)
#the net premium
pi.net <- V.0
```

R-Code 10.2: Calculating the net premium π^{net} in the KLD/AIC-model

The computation yields a net premium of

$$\pi^{\text{net}} = \text{pi.net} = 23.46$$

(rounded to two decimal digits). We used a single core clocked at 2.2GHz of an Intel Xeon (E5-2650 v4). The computation took 36 hours and 20 minutes to complete. We ran this computation seven times simultaneously, once for each model. The computation time was about the same for all models.

The following Table 10.5 lists the results. The third column of the table contains the relative differences (rounded to two decimal digits) when comparing the net premium of the model to the net premium of the reference model MC1, the fourth column lists the additional model complexity compared to MC1 and the fifth column is the premium difference divided by the additional complexity.

model	π^{net}	premium difference	additional complexity	ratio
KLD/AIC	23.46	-3.89%	16.86%	-0.23
L1/AIC	22.50	-7.82%	31.31%	-0.25
L1/BIC	23.37	-4.26%	14.35%	-0.30
MC1	24.41	0.00%	0.00%	
MC2	20.69	-15.24%	496.34%	-0.03
MC3	18.36	-24.78%	3,455.50%	-0.01
MC6	17.55	-28.10%	753,235.60%	0.00

Table 10.5: The net premiums π^{net} in Euro of the seven different models

As we see, the net premiums differ significantly. We can detect a trend: As the model complexity increases, the amount the customer has to pay decreases, i.e. π^{net} decreases. Here it is not the complexity of the complete model that is crucial, but rather the complexity during the contract duration, i.e. at $t = 792, ..., 803$. In the most complex model MC6 the insurance company would price the contract at 17.55 Euro, 28.10% cheaper than in the most expensive MC1-model, where a customer would have to pay 24.41 Euro. At the ages 792, ..., 802 the tree heights of the three tiVLMC-models KLD/AIC, L1/AIC and L1/BIC vary between one and two. This harmonizes with their net premiums arranging themselves in between the net premiums of the MC1- and MC2-model. If we choose e.g. the L1/AIC-model, the higher model precision allows us to price the contract 7.82% cheaper compared to the less precise MC1-model. The embedding full time-discrete Markov process (over the complete time axis) MC3 of L1/AIC would yield a price decrease of 24.78%, but L1/AIC is 96.31% less complex, cf. Table 10.2.
If we take a closer look at the fifth column of Table 10.5, we can see that it is reasonable to accept the additional complexity of the tiVLMC-models because this complexity increase significantly leads to the change in the premium. Adding even more complexity by moving over to MC2, MC3 or even MC6, this effect vanishes. This backs up the fact that a tiVLMC is complex where it is needed to describe the dependency structure, while a full time-discrete Markov process is

just complex everywhere.
The contract we consider has a duration of one year, thus even a percentage difference in the single digit range is of practical relevance. As the contract duration is increased to multiple decades, we expect the differences to grow as they procreate themselves.

10.4.4 Some prospective reserves

In each model we calculate the prospective reserves in each state $x \in M$, i.e. we calculate

$$V_{792}\left(x^6\right), V_{781}\left(x^7\right), ..., V_{803}\left(x^{17}\right). \tag{10.3}$$

R-Code 10.3 is a continuation of R-Code 10.2 and shows how the calculations are carried out for $x = a$ in KLD/AIC.

```
#choose the net premium as the premium
payments <- paymentTree(pi.net)
#the prospective reserves for t.s, ..., t.e-1
V <- NULL
for (t in t.s:(t.e-1)) {
  V <- rbind(V, calculateProspectiveReserve(past=past,
             tau=tau, payments=payments,
             interests=interests))
  past <- c(past, "a")
  tau <- tau[2:length(tau)]
  payments <- payments[2:length(payments)]
  interests <- interests[2:length(interests)]
}
#the prospective reserve for t.e
V <- rbind(V, calculateProspectiveReserve(past=past,
           tau=NULL, payments=payments[2],
           interests=NULL))
```

R-Code 10.3: Calculating the prospective reserve for an active insurance holder

We perform this calculation $6 \cdot 7 = 42$ times, once for each combination of one of the six states and the seven models. As there are no payments in state d, it is not necessary to calculate the prospective reserves for deceased insurance holders. We still do, since it means no additional effort for us. Keeping the computational environment unchanged, all the (simultaneously performed) runs took about 44 hours each. The following two Tables 10.6 and 10.7 list the calculated prospective reserves (10.3) rounded to two decimal digits at the different ages for all models.

		t										
		792	793	794	795	796	797	798	799	800	801	802	803
KLD/AIC	a	0.00	20.50	17.39	13.92	11.60	8.16	6.24	4.34	2.08	1.03	0.31	0.00
	1	5,019.86	4,646.81	4,287.85	3,939.22	3,575.03	3,188.66	2,770.16	2,259.31	1,817.89	1,386.05	928.50	468.00
	2	10,832.24	10,080.78	9,400.36	8,739.02	7,813.60	6,942.68	5,963.66	5,172.27	4,219.52	3,227.64	2,217.89	1,144.00
	3	14,385.35	13,232.39	12,217.50	10,976.69	10,313.50	9,156.73	7,802.51	6,740.90	5,584.93	4,566.05	3,109.21	1,612.00
	$3h$	20,563.49	18,939.86	17,283.76	15,594.54	13,871.53	12,114.06	10,321.44	8,492.97	6,627.93	4,725.58	2,785.20	1,995.00
	d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
L1/AIC	a	0.00	19.53	16.41	13.12	11.01	7.61	5.70	3.89	2.05	0.99	0.31	0.00
	1	5,001.49	4,627.75	4,267.78	3,918.53	3,555.34	3,174.05	2,754.97	2,259.31	1,817.89	1,386.05	928.50	468.00
	2	10,822.03	10,070.08	9,393.78	8,731.97	7,806.13	6,934.97	5,955.62	5,172.27	4,219.52	3,227.64	2,217.89	1,144.00
	3	14,385.17	13,232.21	12,217.31	10,976.49	10,313.39	9,156.72	7,802.50	6,740.90	5,584.93	4,566.05	3,109.21	1,612.00
	$3h$	20,563.49	18,939.86	17,283.76	15,594.54	13,871.53	12,114.06	10,321.44	8,492.97	6,627.93	4,725.58	2,785.20	1,995.00
	d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
L1/BIC	a	0.00	20.40	17.29	13.82	11.50	8.06	6.13	4.34	2.08	1.03	0.31	0.00
	1	5,017.16	4,644.00	4,284.89	3,936.17	3,571.82	3,185.25	2,770.16	2,259.31	1,817.89	1,386.05	928.50	468.00
	2	10,832.20	10,080.74	9,400.31	8,738.97	7,813.56	6,942.68	5,963.66	5,172.27	4,219.52	3,227.64	2,217.89	1,144.00
	3	14,385.35	13,232.39	12,217.50	10,976.69	10,313.50	9,156.73	7,802.51	6,740.90	5,584.93	4,566.05	3,109.21	1,612.00
	$3h$	20,563.49	18,939.86	17,283.76	15,594.54	13,871.53	12,114.06	10,321.44	8,492.97	6,627.93	4,725.58	2,785.20	1,995.00
	d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 10.6: The prospective reserves of an insured remaining in state $x \in M$ over the complete contract duration in the models KLD/AIC, L1/AIC and L1/BIC

		t										
		792	793	794	795	796	797	798	799	800	801	802	803
MC1	a	0.00	21.50	17.62	14.19	11.54	8.11	6.19	4.40	2.14	1.08	0.31	0.00
	1	4,929.31	4,552.84	4,210.96	3,859.92	3,524.36	3,174.42	2,770.19	2,259.35	1,817.89	1,386.05	928.50	468.00
	2	10,740.73	9,984.88	9,344.96	8,703.83	7,776.01	6,903.22	5,922.14	5,172.28	4,219.52	3,227.64	2,217.89	1,144.00
	3	14,384.47	13,231.48	12,216.54	10,975.70	10,312.97	9,156.73	7,802.51	6,740.90	5,584.93	4,566.05	3,109.21	1,612.00
	$3h$	20,563.49	18,939.86	17,283.76	15,594.54	13,871.53	12,114.06	10,321.44	8,492.97	6,627.93	4,725.58	2,785.20	1,995.00
	d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MC2	a	0.00	18.03	15.04	12.01	10.10	7.14	5.37	3.58	1.91	0.89	0.30	0.00
	1	4,954.97	4,631.08	4,268.57	3,899.45	3,535.10	3,156.99	2,723.65	2,245.35	1,820.94	1,388.63	930.90	468.00
	2	11,211.39	10,481.23	9,788.38	8,968.39	8,003.92	7,067.70	6,056.55	5,289.45	4,295.86	3,243.24	2,228.11	1,144.00
	3	15,038.85	13,695.39	12,520.69	11,300.72	10,540.91	9,284.89	7,939.78	6,833.94	5,712.32	4,619.28	3,170.13	1,612.00
	$3h$	20,565.43	18,940.82	17,290.61	15,599.20	13,878.12	12,120.09	10,331.32	8,501.90	6,631.82	4,727.56	2,787.27	1,995.00
	d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MC3	a	0.00	16.41	13.78	10.77	9.27	6.33	4.73	3.26	1.74	0.82	0.27	0.00
	1	4,965.03	4,642.80	4,245.82	3,876.66	3,539.06	3,178.50	2,740.75	2,264.92	1,836.00	1,389.15	931.05	468.00
	2	11,378.83	10,659.91	9,889.20	8,982.05	8,051.23	7,118.39	6,133.64	5,328.55	4,337.59	3,276.66	2,232.13	1,144.00
	3	15,793.81	14,465.45	13,325.44	12,140.42	10,956.65	9,728.38	8,278.71	7,218.45	5,891.87	4,634.11	3,192.39	1,612.00
	$3h$	20,612.86	18,973.13	17,321.86	15,621.54	13,965.82	12,170.01	10,350.40	8,521.51	6,639.93	4,739.28	2,790.21	1,995.00
	d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MC6	a	0.00	13.53	11.25	9.13	7.69	5.98	4.81	2.89	1.84	0.87	0.22	0.00
	1	4,976.67	4,630.23	4,222.95	3,855.68	3,524.82	3,147.67	2,743.20	2,267.22	1,837.08	1,389.31	930.34	468.00
	2	11,697.92	10,878.46	10,012.68	9,151.03	8,181.81	7,223.81	6,221.74	5,330.51	4,348.54	3,290.78	2,248.90	1,144.00
	3	16,405.09	15,088.95	13,991.96	12,627.56	11,293.93	10,097.89	8,655.61	7,432.16	5,936.57	4,726.13	3,192.39	1,612.00
	$3h$	20,660.16	19,023.86	17,400.60	15,676.04	14,030.92	12,223.18	10,384.38	8,530.39	6,650.01	4,751.23	2,799.63	1,995.00
	d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 10.7: The prospective reserves of an insured remaining in state $x \in M$ over the complete contract duration in the models MC1, MC2, MC3 and MC6

It is hard to detect visual differences in the graphs of the prospective reserves of the different models. Therefore we continue as follows: Figure 10.8 shows the prospective reserves (10.3) for each state $x \in M$ for the reference model MC1. Since the basic shapes of the reserves are the same in all seven models, this gives the reader a good first impression. Then we plot the relative differences in percent of the prospective reserve in that state compared to the prospective reserve in the reference model MC1 in the Figures 10.9-10.11. As it makes detecting trends more easy, we connect two consecutive points with a straight line.

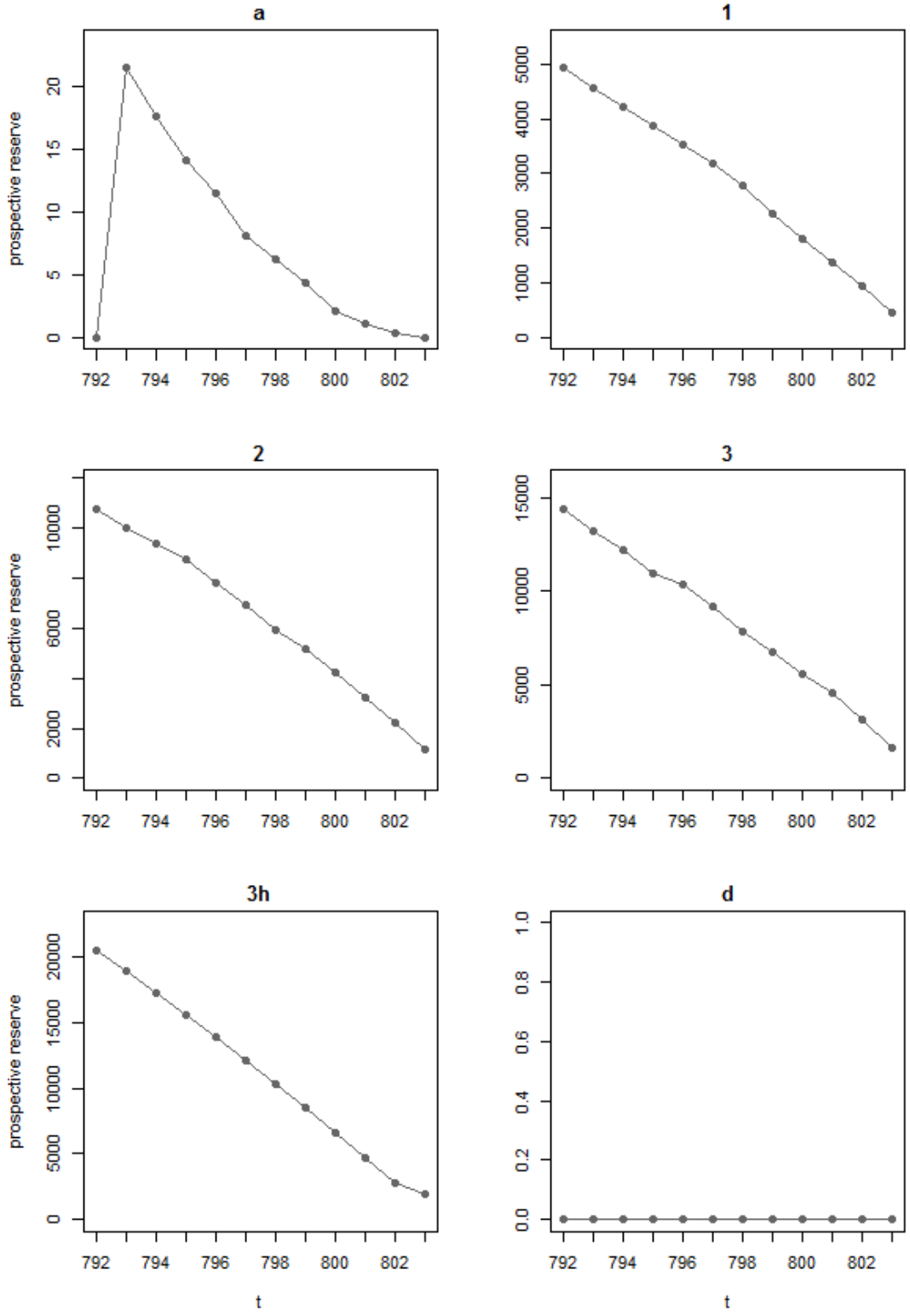

Figure 10.8: The prospective reserves in the MC1-model

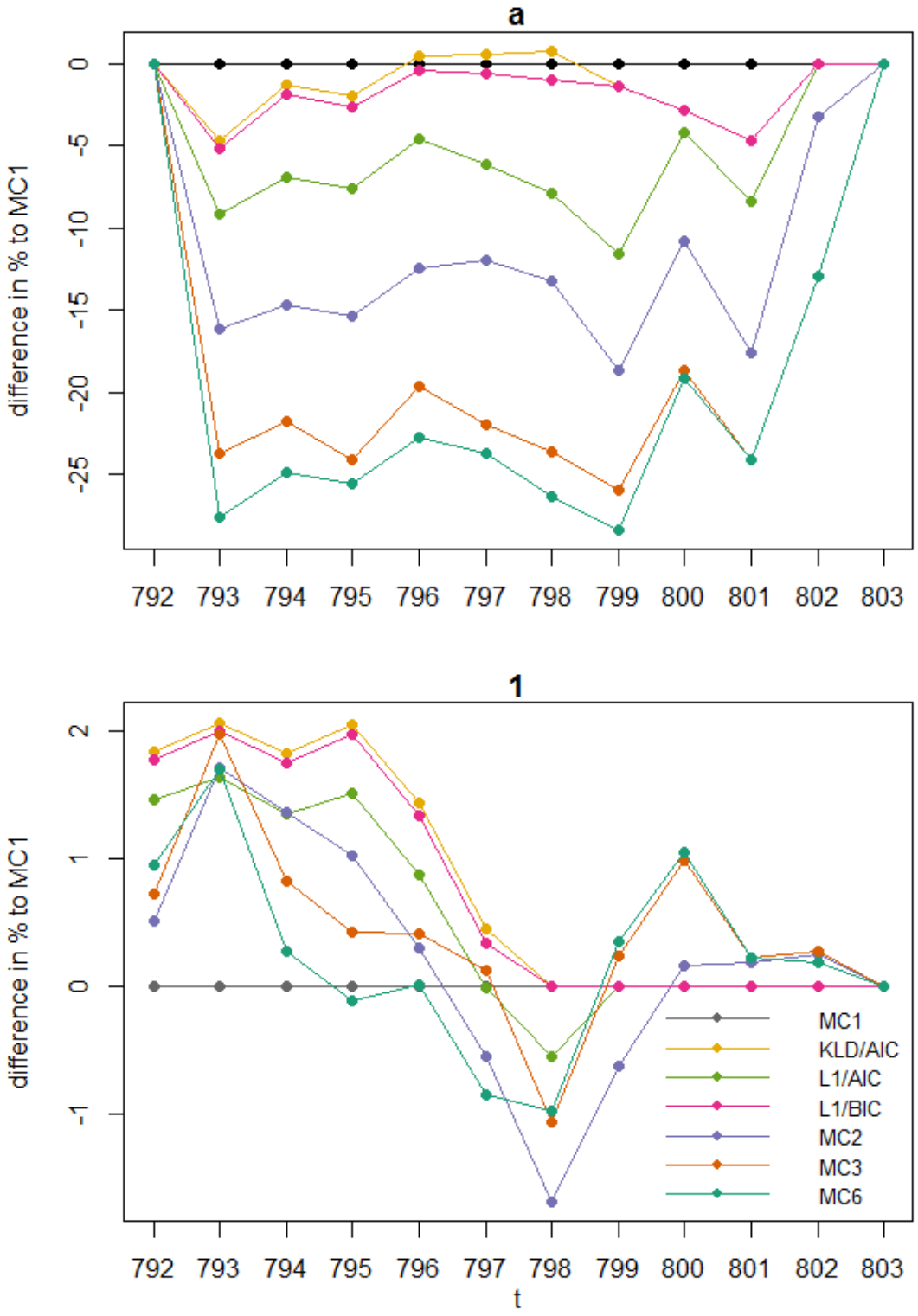

Figure 10.9: The relative differences in the prospective reserves in a and 1 compared to the MC1-model

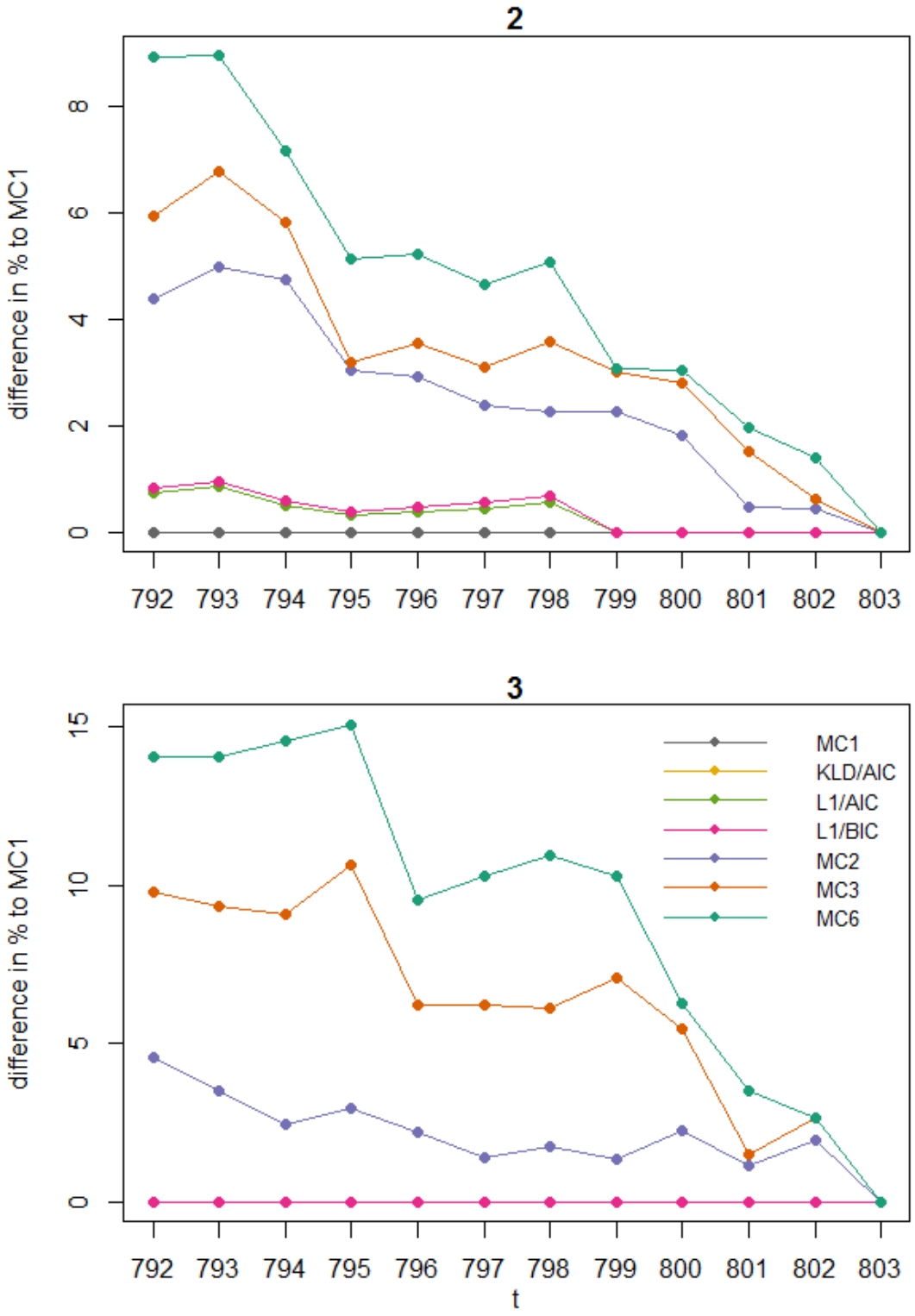

Figure 10.10: The relative differences in the prospective reserves in 2 and 3 compared to the MC1-model

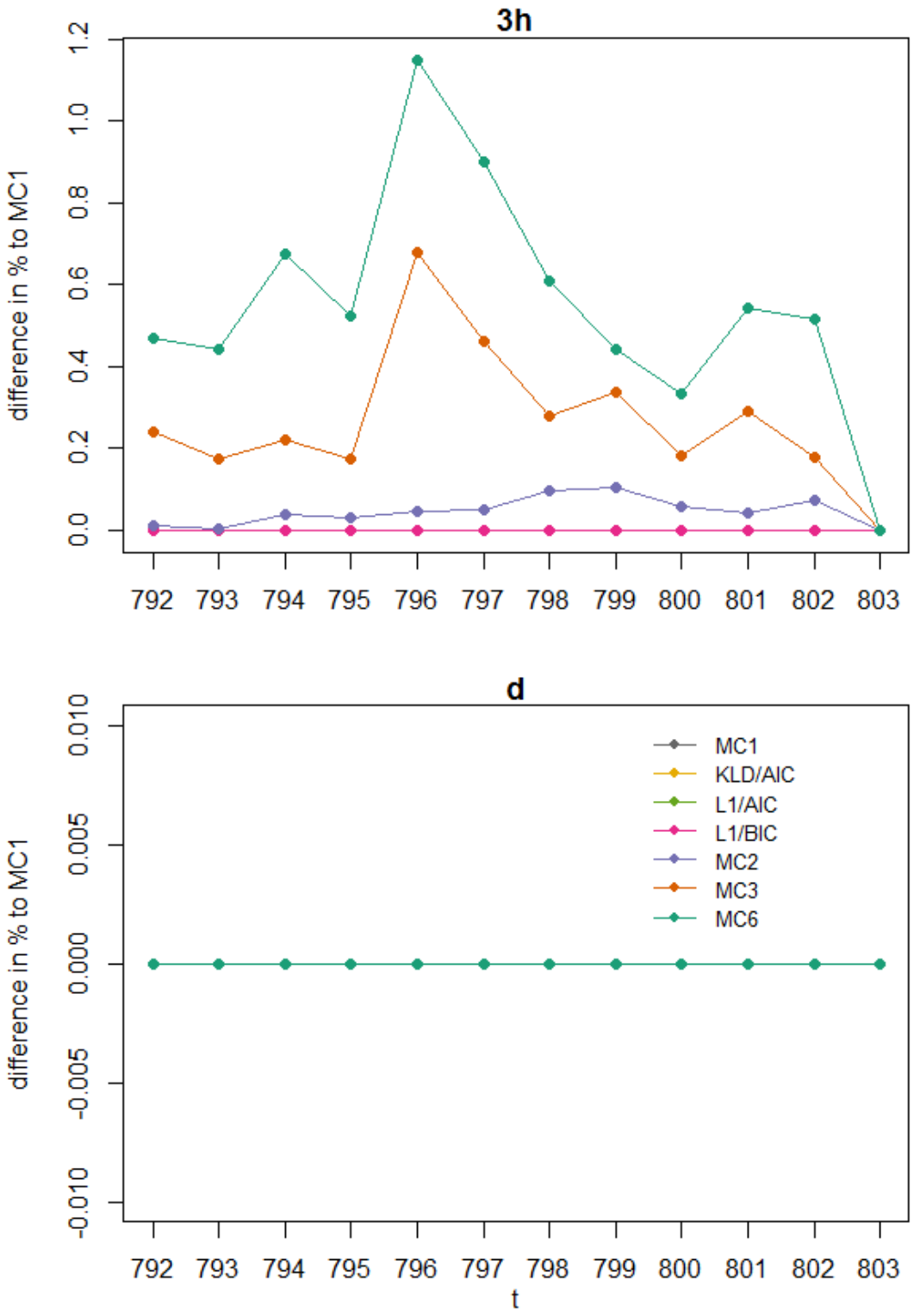

Figure 10.11: The relative differences in the prospective reserves in $3h$ and d compared to the MC1-model

The prospective reserve in state a of the L1/AIC-model at $t = 799$ is 11.59% smaller compared to the MC1-model. This is the largest difference for any of the three tiVLMC-models KLD/AIC, L1/AIC and L1/BIC. The largest difference between all seven models can also be found at $t = 799$: The MC6-model's prospective reserve is 28.41% smaller than the prospective reserve in MC1.
In state 1 the largest difference of a tiVLMC-model is 2.06% and this is also the largest difference among all seven models.
In state 2 the tiVLMC-models maximally differ 0.96%, while the largest overall difference is much larger at 8.94%.
In state 3 or $3h$ there are no differences between the tiVLMC-models and MC1. This is the case because in all of these four models 3 as well as $3h$ are leaves of the context tree of the model. 3 is an internal node of the context tree of the MC6-model and the prospective reserve of the MC6-model in 3 deviates an astonishing 15.05% at $t = 795$ compared to MC1.
The maximal overall difference in state $3h$ is 1.15% for MC6 at $t = 796$.
Obviously there are no differences in state d.
As prospective reserves build the basis for the calculation of a net premium, cf. Definition 9.12, the remark we made after the calculation of the net premiums also applies here: If the contract duration is increased, the differences are expected to grow since they will reproduce themselves over time.
Additionally, if we consider e.g. a contract running for multiple decades, aging will play a more and more important role: The bandwidth of a transition probability $\tilde{P}_t(x \mid w)$ increases as we increase the contract duration, i.e.

$$\left[\min_{t\in\mathscr{T}} \tilde{P}_t\left(x \mid w\right), \max_{t\in\mathscr{T}} \tilde{P}_t\left(x \mid w\right)\right] \subseteq \left[\min_{t\in\mathscr{T}'} \tilde{P}_t\left(x \mid w\right), \max_{t\in\mathscr{T}'} \tilde{P}_t\left(x \mid w\right)\right]$$

holds for $\mathscr{T} \subseteq \mathscr{T}'$. Thus e.g. changes in morbidity due to getting older are of greater impact in long-term contracts.
When we keep the contract duration constant, age still affects the prospective reserves. In order to show how the prospective reserves behave as we change our target group to older customers, assume that we now sell exactly the same contract as before to customers that are $t_s = 864$ months (72 years) old at contract closure and that are thus six years older than our initial target group. Since the following observation is the same for all models, we exemplarily focus on the KLD/AIC-model here.

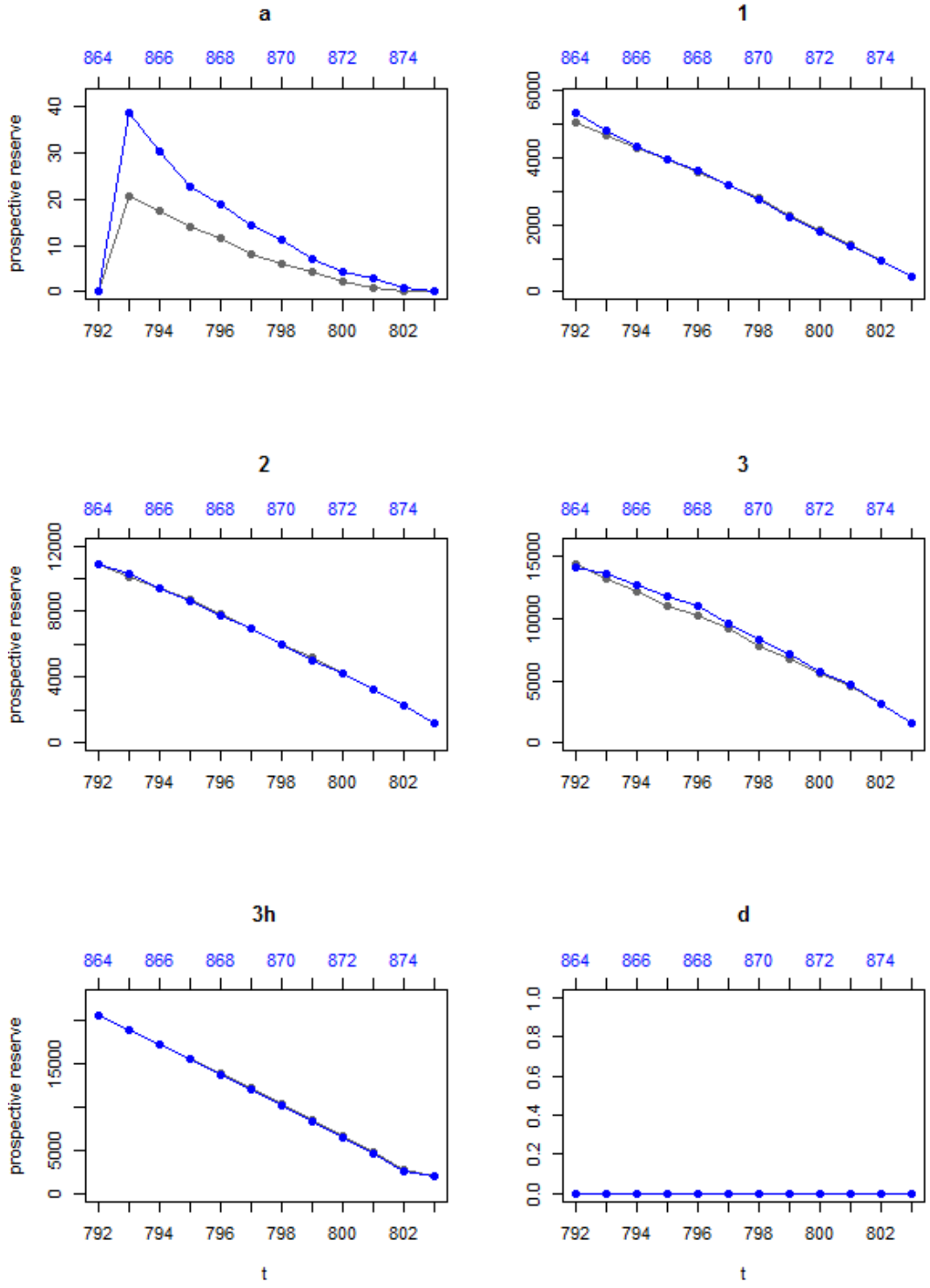

Figure 10.12: The influence of the age t_s at contract closure in KLD/AIC

We calculate

$$V_{864}\left(x^6\right), V_{865}\left(x^7\right), ..., V_{875}\left(x^{17}\right)$$

for all states $x \in M$. Figure 10.12 shows how the higher age at contract closure affects the prospective reserves in each state. The prospective reserves are plotted for both contract start times $t_s = 792$ (black) and $t_s = 864$ (blue).
The insurance holder who is six years older has to pay a net premium of 50.36 Euro at contract closure. This is more than double the net premium that an insurance holder who is 792 months old at contract start has to pay (23.46 Euro). While the prospective reserves basically remain unchanged in the states where the insurance pays out disability benefits, we can see that the prospective reserve in state a is much larger. This fits our intuition: If an insured gets disabled, i.e. transits into 1, 2, 3 or $3h$, the increased age of six years has no significant effect on the future health development. E.g. we do not expect a 72-year-old person to have drastically different chances of e.g. recovering or dying compared to a 66-year-old insurance holder. Thus the corresponding prospective reserves are very similar. But in fact we do expect that an active 72-year-old insurance holder has a much higher probability of getting disabled in the first place compared to an active 66-year-old one. Thus the reserve in state a is larger for the older insurance holder and this causes the increase of the net premium.

11 Interpreting a tiVLMC-model

Because of their high flexibility, tiVLMC can represent complex dependency structures. If we fix a certain time t, visualizing the dependency structure at that time by looking at a context tree is a nice and clear way to interpret the tiVLMC-model. But, when varying the time t, one ends up with an extensive list of context trees, cf. Figures B.1-B.43, and in many cases keeping track of the development of the dependency structure gets quite hard. Here methods which allow an easier interpretation of the dependency structure are needed.
A tiVLMC-model gets easier to interpret and easier to explain to a potential customer if the movement of the estimated context trees over time is reduced, i.e. if the context trees at time points t change less often or change in a more "continuous" fashion. Graphically this translates to smoothing the vertical context tree movement plots, cf. e.g. Figure 10.4.
Within this chapter we will therefore develop different techniques that allow us to smoothen trained tiVLMC-models. In the first subchapter Chapter 11.1 we introduce "break-point-tiVLMC"-models and the concept of supertrees. Using these tools we continue by motivating and constructing "moving-average-tiVLMC"- and "LASSO-tiVLMC"-models, two different smoothing approaches.

11.1 Break-point-tiVLMC

An important concept for the simplification of tiVLMC-models is what we call the supertree:

Definition 11.1. For a set of times $\mathscr{T} \subseteq \{1, ..., T-1\}$ and trees $\{\tau_r : r \in \mathscr{T}\}$ let $\tau_{\mathscr{T}}^{\text{super}}$ be the unique tree for which

$$\tau_r \preccurlyeq \tau_{\mathscr{T}}^{\text{super}}$$

holds for all $r \in \mathscr{T}$ and for any other $\mathcal{T}$ with $\tau_r \preccurlyeq \mathcal{T}$ for all $r \in \mathscr{T}$ it is implied that

$$\tau_{\mathscr{T}}^{\text{super}} \preccurlyeq \mathcal{T}.$$

If $\mathscr{T} = \{l, ..., k\}$ for two times $1 \leq l \leq k \leq T-1$, we also write $\tau_{l:k}^{\text{super}} := \tau_{\mathscr{T}}^{\text{super}}$.

In words, $\tau_{l:k}^{\text{super}}$ is the smallest tree including (w.r.t. the relation $\preccurlyeq$ defined in Definition 4.6) all context trees $\tau_l, ..., \tau_k$.

Example 11.2. The supertree $\tilde{\tau}_{727:885}^{\text{super}}$ of the AIC-tuned tiVLMC fitted using the Kullback-Leibler divergence in Chapter 10.3, i.e. the smallest tree that includes all context tree estimates $\tilde{\tau}_{727}, ..., \tilde{\tau}_{885}$, is plotted in the following Figure 11.1.

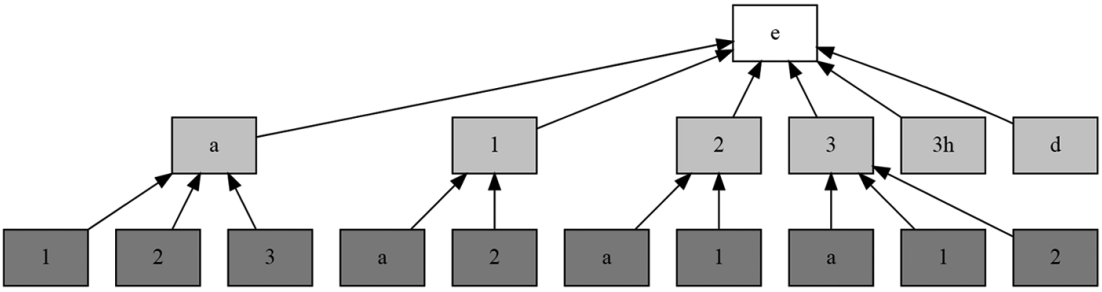

Figure 11.1: The supertree of the AIC-tuned tiVLMC-model trained on the data.signal data set using measure="KLD"

It happens that this is also the supertree of the BIC-tuned tiVLMC using the Kullback-Leibler divergence or using the L^1-norm.

Since the supertree $\tau_{1:T-1}^{\text{super}}$ includes every context tree at time $t = 1, ..., T-1$, it is a good visualization of the overall dependency structure of a data-generating tiVLMC. However, at a fixed t, the supertree might be considerably larger than the true context tree at that time and thus it might be a harsh simplification.
What one can observe in practice is that in some cases the context trees do not vary with every step from time t to time $t+1$ but rather only vary at some "break points". If the context trees vary, they often preserve certain properties, e.g. the height of the context tree might stay the same or they only differ at certain branches. In such cases there is a good chance that trading a part of the model precision for a better interpretability of the model fit is beneficial to the practitioner.
We can trade model precision for interpretability by limiting the number of context tree changes to a chosen integer $0 \leq k \leq T-2$. Between the break points we use the supertree of the estimated context trees as a replacement for the estimated context tree itself, since it is the smallest tree capable of representing the complete dependency structures of the tiVLMC on the time segments defined by the break points. For $k = 0$ one ends up with only one context tree, the supertree

$\tilde{\tau}^{\text{super}}_{1:T-1}$ for the complete time axis. For $k = T-2$ one ends up with the originally estimated context tree $\tilde{\tau}$, since $\tilde{\tau}^{\text{super}}_{l:l} = \tilde{\tau}_l$, $1 \leq l \leq T-1$. For $1 \leq k \leq T-3$, e.g. we can calculate the break points

$$1 \leq b_1^{\text{AIC}} < ... < b_k^{\text{AIC}} \leq T-2$$

or

$$1 \leq b_1^{\text{BIC}} < ... < b_k^{\text{BIC}} \leq T-2$$

by solving

$$\left(b_1^{\text{AIC}}, ..., b_k^{\text{AIC}}\right) = \underset{\substack{(b_1,...,b_k):\\ 1\leq b_1<...<b_k\leq T-2}}{\arg\min} \text{AIC}\left(C_{\text{AIC}}; b_1, ..., b_k\right) \tag{11.1}$$

or

$$\left(b_1^{\text{BIC}}, ..., b_k^{\text{BIC}}\right) = \underset{\substack{(b_1,...,b_k):\\ 1\leq b_1<...<b_k\leq T-2}}{\arg\min} \text{BIC}\left(C_{\text{BIC}}; b_1, ..., b_k\right) \tag{11.2}$$

respectively. Recall that

$$\begin{aligned}
\text{AIC}\left(C_{\text{AIC}}; b_1, ..., b_k\right) =& -2\log L\left(\tilde{\tau}^{\text{super}}(b_1, ..., b_k)\right) \\
&+ 2(m-1)\sum_{t=1}^{T-1} |\tilde{\tau}^{\text{super}}(b_1, ..., b_k)_t|, \\
\text{BIC}\left(C_{\text{BIC}}; b_1, ..., b_k\right) =& -2\log L\left(\tilde{\tau}^{\text{super}}(b_1, ..., b_k)\right) \\
&+ \log n \cdot (m-1)\sum_{t=1}^{T-1} |\tilde{\tau}^{\text{super}}(b_1, ..., b_k)_t|
\end{aligned}$$

as proposed in Chapter 6.3. Instead of evaluating the information criteria using the usual estimated context tree

$$\tilde{\tau} = (\tilde{\tau}_1, ..., \tilde{\tau}_{T-1})$$

output by Algorithm 5.6, the segment-wise supertree $\tilde{\tau}^{\text{super}}(b_1, ..., b_k)$ is used:

Definition 11.3. For $1 \leq k \leq T-3$, break points $1 \leq b_1 < ... < b_k \leq T-2$ and a context tree $\tau = (\tau_t)_{1\leq t\leq T-1}$ we define the segment-wise supertree by

$$\tau^{\text{super}}(b_1,...,b_k) := \Big(\underbrace{\tau^{\text{super}}_{1:b_1},...,\tau^{\text{super}}_{1:b_1}}_{b_1\text{-times}}, \underbrace{\tau^{\text{super}}_{b_1+1:b_2},...,\tau^{\text{super}}_{b_1+1:b_2}}_{(b_2-b_1)\text{-times}}, ...,$$
$$\underbrace{\tau^{\text{super}}_{b_{k-1}+1:b_k},...,\tau^{\text{super}}_{b_{k-1}+1:b_k}}_{(b_k-b_{k-1})\text{-times}}, \underbrace{\tau^{\text{super}}_{b_k+1:T-1},...,\tau^{\text{super}}_{b_k+1:T-1}}_{(T-1-b_k)\text{-times}} \Big).$$

Switching over to the sequence $\tilde{\tau}^{\text{super}}\left(b_1^{\text{AIC}},...,b_k^{\text{AIC}}\right)$ or $\tilde{\tau}^{\text{super}}\left(b_1^{\text{BIC}},...,b_k^{\text{BIC}}\right)$ of supertrees, which coordinate-wise always include the estimated context tree, the likelihood of the model increases. Since the supertrees are larger than the estimated context trees, the model complexity also increases.
The relation $\preccurlyeq$ implies a partial order onto the set of tree sequences of the same length:

Definition 11.4. For $\mathscr{T} \subseteq \{1,...,T-1\}$ we call the sequence $\tau = (\tau_t)_{t\in\mathscr{T}}$ of sets smaller than or equal to the sequence $\mathcal{T} = (\mathcal{T}_t)_{t\in\mathscr{T}}$ of sets, written $\tau \preccurlyeq \mathcal{T}$, if

$$\tau_t \preccurlyeq \mathcal{T}_t$$

holds for all $t \in \mathscr{T}$.

Note that we do not introduce a new symbol for the relation between two tree sequences, since it is always clear from the context whether single trees at a fixed time or complete sequences are compared.
Using the segment-wise supertree instead of the estimated context tree $\tilde{\tau}$ output by Algorithm 5.6 obviously always yields a tree sequence that includes the true data-generating context tree.

Theorem 11.5. *Let $X = (X_t)_{1\le t\le T} \in \mathcal{P}^{0:T-1}$ with $T \ge 2$ be a tiVLMC with state space M of cardinality $2 \le m < \infty$ and (unknown) context tree τ. If $X^1_{1:T} = x^1_{1:T},...,X^n_{1:T} = x^n_{1:T}$ are n independent observations of X, censored by censoring variables $J^1_{1:T} = j^1_{1:T},...,J^n_{1:T} = j^n_{1:T}$ that satisfy (H.1)-(H.5), and $\tilde{\tau}$ is the output of Algorithm 5.6,*

$$\lim_{n\to\infty} \mathbb{P}\left(\tau_t \preccurlyeq \tilde{\tau}^{\text{super}}(b_1,...,b_k)_t\right) = 1$$

holds for all $1 \leq t \leq T-1$ *and for arbitrary break points* $1 \leq b_1 < ... < b_k \leq T-2$, $0 \leq k \leq T-2$.

Proof of Theorem 11.5. Choose arbitrary break points $1 \leq b_1 < ... < b_k \leq T-2$, $0 \leq k \leq T-2$. Fix a time t, $1 \leq t \leq T-1$. Since the break points $b_1, ..., b_k$ induce the partition

$$\{1, ..., b_1\}, \{b_1 + 1, ..., b_2\}, ..., \{b_{k-1} + 1, ..., b_k\}, \{b_k, ..., T-1\}$$

of $\{1, ..., T-1\}$, there exists a unique index $0 \leq k_0 \leq T-2$ such that

$$b_{k_0} + 1 \leq t \leq b_{k_0+1},$$

where we define $b_0 := 0$ and $b_{T-1} := T-1$. Then it holds per definition that

$$\tilde{\tau}_t \preccurlyeq \tilde{\tau}^{\text{super}}_{b_{k_0}+1:b_{k_0+1}}.$$

Therefore, by applying Corollary 5.9,

$$\lim_{n\to\infty} \mathbb{P}\left(\tau_t = \tilde{\tau}_t \preccurlyeq \tilde{\tau}^{\text{super}}_{b_{k_0}+1:b_{k_0+1}}\right) = \lim_{n\to\infty} \mathbb{P}\left(\tau_t \preccurlyeq \tilde{\tau}^{\text{super}}(b_1, ..., b_k)_t\right) = 1.$$

Since t was arbitrary, this holds for all t. □

For fixed break points $1 \leq b_1 < ... < b_k \leq T-2$, $0 \leq k \leq T-2$, the estimated segment-wise supertree $\tilde{\tau}^{\text{super}}(b_1, ..., b_k)$ is the smallest tree sequence asymptotically including the true context tree. I.e. any other estimator $\mathcal{T}$ is either underestimating or unnecessarily large. We put this on record:

Proposition 11.6. *Let* $X = (X_t)_{1\leq t\leq T} \in \mathcal{P}^{0:T-1}$ *with* $T \geq 2$ *be a tiVLMC with state space* M *of cardinality* $2 \leq m < \infty$ *and (unknown) context tree* τ*. If* $X^1_{1:T} = x^1_{1:T}, ..., X^n_{1:T} = x^n_{1:T}$ *are* n *independent observations of* X*, censored by censoring variables* $J^1_{1:T} = j^1_{1:T}, ..., J^n_{1:T} = j^n_{1:T}$ *that satisfy (H.1)-(H.5), and* $\tilde{\tau}$ *is the output of Algorithm 5.6,*

$$\lim_{n\to\infty} \mathbb{P}\left(\tau_t \preccurlyeq \mathcal{T}_t\right) = 1$$

holds for all $1 \leq t \leq T-1$ *and for break points* $1 \leq b_1 < ... < b_k \leq T-2$, $0 \leq k \leq T-2$ *if and only if*

$$\tilde{\tau}^{\text{super}}(b_1, ..., b_k) \preccurlyeq \mathcal{T}.$$

Proof of Proposition 11.6.
"$\Leftarrow$":
By Theorem 11.5 it is implied that

$$\lim_{n\to\infty} \mathbb{P}\left(\tau_t \preccurlyeq \tilde{\tau}^{\text{super}}(b_1, ..., b_k)_t\right) = 1$$

holds. If $\tilde{\tau}^{\text{super}}(b_1, ..., b_k) \preccurlyeq \mathcal{T}$, we therefore have

$$1 \geq \lim_{n\to\infty} \mathbb{P}\left(\tau_t \preccurlyeq \tilde{\tau}^{\text{super}}(b_1, ..., b_k)_t \preccurlyeq \mathcal{T}_t\right) = 1$$

and thus

$$\lim_{n\to\infty} \mathbb{P}\left(\tau_t \preccurlyeq \mathcal{T}_t\right) = 1.$$

"$\Rightarrow$":
We prove this claim by contraposition. If $\tilde{\tau}^{\text{super}}(b_1, ..., b_k) \preccurlyeq \mathcal{T}$ does not hold, per definition there has to exist at least one time $1 \leq t_0 \leq T-2$ such that

$$\tilde{\tau}^{\text{super}}(b_1, ..., b_k)_{t_0} \preccurlyeq \mathcal{T}_{t_0}$$

is violated. Since the break points $b_1, ..., b_k$ induce the partition

$$\{1, ..., b_1\}, \{b_1+1, ..., b_2\}, ..., \{b_{k-1}+1, ..., b_k\}, \{b_k, ..., T-1\}$$

of $\{1, ..., T-1\}$, there exists a unique index $0 \leq k_0 \leq T-2$ such that

$$b_{k_0} + 1 \leq t_0 \leq b_{k_0+1},$$

where we again define $b_0 := 0$ and $b_{T-1} := T-1$. Since $\tilde{\tau}^{\text{super}}(b_1, ..., b_k)$ does not change within the time interval from $b_{k_0}+1$ to b_{k_0+1} and since the estimated supertree $\tilde{\tau}^{\text{super}}_{b_{k_0}+1:b_{k_0+1}}$ is the smallest tree including the estimated context trees $\tilde{\tau}_{b_{k_0}+1}, ..., \tilde{\tau}_{b_{k_0+1}}$, there must exist a time $b_{k_0}+1 \leq t_1 \leq b_{k_0+1}$ for which

$$\tilde{\tau}_{t_1} \preccurlyeq \mathcal{T}_{t_1}$$

is violated, i.e. particularly $\tilde{\tau}_{t_1} \neq \mathcal{T}_{t_1}$ holds. Corollary 5.9 implies

$$\lim_{n\to\infty} \mathbb{P}\left(\tilde{\tau}_{t_1} = \tau_{t_1}\right) = 1$$

and thus

$$\lim_{n\to\infty} \mathbb{P}\left(\tilde{\tau}_{t_1} \preccurlyeq \mathcal{T}_{t_1}\right) = 0$$

which completes the proof. □

Summarized, Theorem 11.5 and Proposition 11.6 prove that the segment-wise supertree is the smallest tree sequence among all tree sequences which only changes its tree k times at the given break points while being capable of displaying the complete dependency structure of the underlying tiVLMC. This justifies the choice of the segment-wise supertree for simplification purposes.
Note that, since Algorithm 5.6 outputs a consistent estimator for every cutoff value $C > 0$, Corollary 11.5 and Proposition 11.6 do also not require any further restrictions on the cutoff. In particular

$$\lim_{n\to\infty} \mathbb{P}\left(\tau_t \preccurlyeq \tilde{\tau}^{\text{super}}\left(b_1^{\text{AIC}}, ..., b_k^{\text{AIC}}\right)_t\right) = 1$$

and

$$\lim_{n\to\infty} \mathbb{P}\left(\tau_t \preccurlyeq \tilde{\tau}^{\text{super}}\left(b_1^{\text{BIC}}, ..., b_k^{\text{BIC}}\right)_t\right) = 1$$

hold for $1 \leq t \leq T-1$ and this justifies the approach presented in (11.1) and (11.2) respectively.

11.2 Moving-average-tiVLMC

Instead of calculating the k optimal (with regards to e.g. AIC or BIC as proposed in above Chapter 11.1) time points for a given k where the tiVLMC-fit is allowed to change its context tree, one can also consider different approaches on how to limit the movement of the estimated context tree in order to simplify interpretation. In this chapter we are going to construct the concept of "moving-average-tiVLMC" as an alternative to break-point-tiVLMC.
The idea of moving-average-tiVLMC is to modify Algorithm 5.6 in order to step away from the "pointwise" setup only considering one time point t within the pruning decision at step two of the algorithm and rather include more than one time point into the decision process. This can be done via a generalization of the

censored time-sensitive pruning measure, cf. Definition 5.5.

Definition 11.7. For $1 \leq t \leq T$, $wu \in M^{1:t}$, where $u \in M$ and $\tilde{N}_{r+}(w) > 0$ for $r \in \mathscr{T}_t$, we define the weighted time-sensitive pruning measure

$$\dot{\Delta}_t(wu) := \sum_{r \in \mathscr{T}_t} g_r \tilde{\Delta}_r(wu),$$

where $\mathscr{T}_t \subseteq \{1, ..., t\}$ is a set of time points with $t \in \mathscr{T}_t$ and $(g_r)_{r \in \mathscr{T}_t} \in \mathbb{R}_{>0}^{|\mathscr{T}_t|}$ are positive weights.

For $\mathscr{T}_t = \{t\}$ and $g_t = 1$ the weighted time-sensitive pruning measure $\dot{\Delta}_t(wu)$ equals the censored time-sensitive pruning measure $\tilde{\Delta}_t(wu)$.
Definition 11.7 enables us to include more than only the current time t into the pruning decision. Thus we can account for the intuitive idea that not only the information at time t, but also the information at other (recent) times is beneficial knowledge for the pruning decision. By the choice of weights e.g. one can calibrate the model fitting in a way that time points farther away in the past are less emphasised than more recent times. This concept has some characteristics in common with the statistical principle known under "borrowing strength" that was originated by John W. Tukey, cf. Wilder (2002, page 196). If one tries to estimate the parameters of a distribution, an example of borrowing strength is the following: Prior knowledge of one parameter will be beneficial to the purpose of estimating the others. The borrowing strength principle is most common in the mathematical discipline of empirical Bayes estimation, e.g. cf. Morris (1983) and Consonni and Veronese (1995).
Another very important motivation for Definition 11.7 and for moving-average-tiVLMC in general is that they allow us to include external information. E.g. assume that a legislator issues a new regulation on data privacy that requires local companies not to consider and/or delete any information about their customers older than a fixed threshold. Now we could account for such regulations by only considering times up to this threshold in our weighted-time-sensitive pruning measure. In the presence of the General Data Protection Regulation (GDPR) issued by the European Parliament in 2016, cf. European Parliament (2016), such considerations are of high relevance. E.g. researchers have just recently started to commit themselves to the question of which challenges the life insurance industry will have to face in an environment of (deliberately) non-complete information, cf. Christiansen and Furrer (2020).

Now we present the fitting algorithm for moving-average-tiVLMC that is based on Definition 11.7:

Algorithm 11.8.
Input: $X^1_{1:T} = x^1_{1:T}, ..., X^n_{1:T} = x^n_{1:T}$, $J^1_{1:T} = j^1_{1:T}, ..., J^n_{1:T} = j^n_{1:T}$, *a cutoff* $C > 0$, *an integer* $n_{\min} \in \mathbb{N}$, *a time point set* $\mathscr{T}_t \subseteq \{1, ..., t\}$ *with* $t \in \mathscr{T}_t$ *and a vector of weights* $(g_r)_{r \in \mathscr{T}_t}$ *for every* $1 \leq t \leq T-1$

For $t = 1, ..., T-1$ *do:*

Step 1: Tree growing
Construct the maximal terminal node context tree $\tau^t_{\max}$ *such that*

$$w \in \tau^t_{\max} \Rightarrow \tilde{N}_{t+}(w) \geq n_{\min}$$

and

$$\forall \tau^t_t \text{ with } w \in \tau^t_t \Rightarrow \tilde{N}_{t+}(w) \geq n_{\min} \text{ it holds that } \tau^t_t \preccurlyeq \tau^t_{\max}.$$

Set $\tau^t_{(0)} := \tau^t_{\max}$.

Step 2: Leaf pruning
Examine every element $wu \in \tau^t_{(0)}$, $u \in M$, *as follows:*
Prune wu *down to* w *if*

$$\dot{\Delta}_t(wu) < C\sqrt{n \log n},$$

else do nothing.
This yields a smaller or equally sized tree $\tau_{(1)} \preccurlyeq \tau^t_{(0)}$.

Step 3: Stopping criterion
Repeat Step 2 with $\tau^t_{(i)}$ *instead of* $\tau^t_{(i-1)}$ *until* $\tau_{(i+1)}$ *and* $\tau^t_{(i)}$ *are equally sized. Denote this maximally pruned context tree* $\dot{\tau}_t$.

Output: $\dot{\tau} := (\dot{\tau}_t)_{1 \leq t \leq T-1}$

Mathematically, the set $\mathscr{T}_t$ and the weights $\{g_r\}_{r \in \mathscr{T}_t}$ in combination form a smoothing kernel at a given time point t. Then the shape of the kernel is taken into account by calculating the weighted time-sensitive pruning measure as an

average of the censored time-sensitive pruning measure. As the time passes by, this averaging procedure/the kernel "moves". This is where the name moving-average-tiVLMC has its origin. E.g. one could use a triangle as the kernel and choose $\mathscr{T}_t = \{t-2, t-1, t\}$ and $g_{t-2} = \frac{1}{3}$, $g_{t-1} = \frac{2}{3}$, $g_t = 1$ as the weights. One can also include future time points into the smoothing and use a discretized Gaussian-bell-shape as the kernel. We are going to apply different combinations of weights and time point sets to real life data in the following Chapter 12.
We also want to mention that the time point sets and corresponding weights imply a pointwise nesting property: When using the same time point set, smaller weights yield a smaller weighted time-sensitive pruning measure and therefore yield to more pruning in step two of above Algorithm 11.8. Thus the estimated context tree calculated with smaller weights will be included in the larger estimated context tree that was obtained by using bigger weights.

Corollary 11.9. *For a fixed time t, $1 \leq t \leq T-1$, let $\dot{\tau}_t$ be the estimated context tree at t output by Algorithm 11.8 using the time point set $\mathscr{T}_t$ and weights $(g_r)_{r\in\mathscr{T}_t}$ and let $\dot{\tau}'_t$ be the estimated context tree at t output using the same time point set but different weights $(g'_r)_{r\in\mathscr{T}_t}$.*
If $g_r \leq g'_r$ for all $r \in \mathscr{T}_t$, it holds that $\dot{\tau}_t \preccurlyeq \dot{\tau}'_t$.

A moving-average-tiVLMC-fit always asymptotically contains the true model. But, as we will see in the following and as it is also the case for break-point-tiVLMC, moving-average-tiVLMC overestimate the true context trees in certain situations.

Theorem 11.10. *Let $X = (X_t)_{1\leq t\leq T} \in \mathcal{P}^{0:T-1}$ with $T \geq 2$ be a tiVLMC with state space M of cardinality $2 \leq m < \infty$ and (unknown) context tree τ. If $X^1_{1:T} = x^1_{1:T}, ..., X^n_{1:T} = x^n_{1:T}$ are n independent observations of X, censored by censoring variables $J^1_{1:T} = j^1_{1:T}, ..., J^n_{1:T} = j^n_{1:T}$ that satisfy (H.1)-(H.5), and $\dot{\tau}$ is the output of Algorithm 11.8,*

$$\lim_{n\to\infty} \mathbb{P}\left(\tau_t \preccurlyeq \dot{\tau}_t\right) = 1$$

holds for all $1 \leq t \leq T-1$.

Proof of Theorem 11.10. Fix t. To prove the claim, we have to show that the probability of underestimating the true context tree τ_t at t vanishes as the observation size n diverges to infinity. Underestimation takes place if and only if there exists a leaf $wu \in \tau_t$, $u \in M$, of the true context tree which is not part of the estimated context tree, i.e. $w \in \dot{\tau}_t^{\mathrm{t}}$. As we have already proven, the pointwise probability in t of underestimation goes to zero as $n \to \infty$, cf. Corollary 5.9. Hence there exists a threshold integer $n_0 \in \mathbb{N}$ such that

$$\tilde{\Delta}_t(wu) \geq C\sqrt{n \log n}$$

holds for all $n \geq n_0$. Then there must also exist a threshold integer $n_1 \in \mathbb{N}_0$ such that

$$g_t \tilde{\Delta}_t(wu) \geq C\sqrt{n \log n} \tag{11.3}$$

for all $n \geq n_1$. Additionally, the frequency counter $\tilde{N}_{r+}(wu)$ as well as the Kullback-Leibler divergence $\mathrm{D}\left[\tilde{P}_r(\cdot|wu)||\tilde{P}_r(\cdot|w)\right]$ and the weight g_r are non-negative for arbitrary $1 \leq r \leq T-1$ and thus

$$g_r \tilde{\Delta}_r(wu) \geq 0$$

holds. Combining this yields

$$\dot{\Delta}_t(wu) = g_t \tilde{\Delta}_t(wu) + \sum_{r \in \mathscr{T}_t \setminus \{t\}} g_r \tilde{\Delta}_r(wu) \geq g_t \tilde{\Delta}_t(wu) \geq C\sqrt{n \log n} \tag{11.4}$$

for all $n \geq n_1$ and therefore the probability of underestimation converges to zero with increasing observation size. □

However, as it is the case for break-point-tiVLMC-models, cf. Theorem 11.5, one cannot generally ensure that a moving-average-tiVLMC-model does not overfit. More precisely: We will see that the context tree estimator output by Algorithm 11.8 overfits at the time t if and only if the true context tree τ_t at t is not the supertree $\tau_{\mathscr{T}_t}^{\mathrm{super}}$ of the trees $\{\tau_r : r \in \mathscr{T}_t\}$.

Theorem 11.11. *Let $X = (X_t)_{1 \leq t \leq T} \in \mathcal{P}^{0:T-1}$ with $T \geq 2$ be a tiVLMC with state space M of cardinality $2 \leq m < \infty$ and (unknown) context tree τ. If $X_{1:T}^1 = x_{1:T}^1, ..., X_{1:T}^n = x_{1:T}^n$ are n independent observations of X, censored by*

censoring variables $J^1_{1:T} = j^1_{1:T}, ..., J^n_{1:T} = j^n_{1:T}$ *that satisfy (H.1)-(H.5), and* $\dot{\tau}$ *is the output of Algorithm 11.8,*

$$\lim_{n\to\infty} \mathbb{P}\left(\dot{\tau}_r \preccurlyeq \tau_r\right) = 1$$

holds for t, $1 \leq t \leq T-1$, *if and only if*

$$\tau^{\text{super}}_{\mathscr{T}_t} = \tau_t.$$

Proof of Theorem 11.11.
"$\Rightarrow$":
Fix t. We prove the claim via contraposition. Let $\tau^{\text{super}}_{\mathscr{T}_t} \neq \tau_t$. Since $t \in \mathscr{T}_t$, it holds by Definition 11.1 that

$$\tau_t \preccurlyeq \tau^{\text{super}}_{\mathscr{T}_t}.$$

Thus there exists (at least) one tree τ_{t_0} with $t_0 \in \mathscr{T}_t \backslash \{t\}$ that possesses (at least) one leaf $wu \in \tau^{\mathsf{t}}_{t_0}$, $u \in M^{1:t}$ that is not a leaf of τ_t:

$$w \in \tau^{\mathsf{t}}_t.$$

Now the consistency result Corollary 5.9 implies that eventually

$$\dot{\Delta}_t(wu) \geq g_{t_0}\tilde{\Delta}_{t_0}(wu) \geq C\sqrt{n \log n}$$

as $n \to \infty$ (as in (11.4)), thus $wu \in \dot{\tau}_t$. Hence

$$\lim_{n\to\infty} \mathbb{P}\left(\dot{\tau}_t \preccurlyeq \tau_t\right) = 0.$$

"$\Leftarrow$":
Fix t. Let $w \in \tau^{\mathsf{t}}_t$ be a leaf of τ_t and $u \in M$. Then $wu \notin \tau^{\mathsf{t}}_t$ and thus especially

$$wu \notin \tau^{\text{super}}_{\mathscr{T}_t}.$$

Therefore it follows that

$$wu \notin \tau^{\mathsf{t}}_r$$

for all $r \in \mathscr{T}_t$. Hence the consistency result Corollary 5.9 states that

$$\tilde{\Delta}_r(wu) = \mathcal{O}\left(\sqrt{n \log\log n}\right)$$

for all $r \in \mathscr{T}_t$ and therefore

$$\begin{aligned}\dot{\Delta}_t(wu) &= \mathcal{O}\left(|\mathscr{T}_t|\sqrt{n\log\log n}\right)\\ &= \mathcal{O}\left(\sqrt{n\log\log n}\right).\end{aligned}$$

It immediately follows that the probability of $wu \in \dot{\tau}_t^{\dagger}$ in fact does vanish as $n \to \infty$ by copying the arguments of the proof of Theorem 5.8.
This completes the proof. □

In R we can fit moving-average-tiVLMC-models using the estimateMovingAverageTau-function displayed in R-Code 11.1.
This function is very similar to the estimateTau-function presented in R-Code 7.1 in Chapter 7.1. The only key difference is that instead of the censored time-sensitive pruning measure the weighted time-sensitive pruning measure has to be calculated, cf. Definitions 5.5 and 11.7. To do so, at a fixed time t, we aggregate the criticalCutoff-attributes of each node while taking the time point set $\mathscr{T}_t$ and the corresponding vector of weights $(g_r)_{r\in\mathscr{T}_t}$ into account.
The user inputs the time point sets as a list of vectors via the time.sets-input and the weights as a list of vectors via the weight.sets-input. We exemplarily demonstrate the handling of estimateMovingAverageTau in Chapter 12.3, where we fit different moving-average-tiVLMC-models to a real life data set documenting joint life insurance contracts.

```
estimateMovingAverageTau <- function(data, ties=NULL, time.sets,
                                     weight.sets, from=NULL, to=NULL,
                                     nmin=2, max.length=NULL,
                                     cutoff, measure, log.like=F) {
  #extracts the state space from the data, NA aka censoring is
  #not considered a state
  alphabet <- na.omit(unique(c(data)))
  #throw an error if one state is labled "e"
  if ("e" %in% alphabet) {
    stop("e is not an allowed state name as it denotes the empty
         string! Rename e within the data.")
  }
  #ties is an optional argument with default value NULL
```

```
#throw an error if the frequencies of the observations do not
#match the number of the observations
if ((length(ties) != dim(data)[1]) & (!is.null(ties))) {
  stop("length(ties) and dim(data)[1] have to be equal!")
}
#from is an optional argument with default value one
if (is.null(from)) from <- 1
#to is an optional argument with the latest time possible as
#default
if (is.null(to)) to <- dim(data)[2]-1
#in this list the tau at t will be saved
tau <- list()
#number of observations
nobs <- dim(data)[1]
#calculates tau.max at all time points by calling the
#estimateTau-function with calculate.tau=F
tau.max <- estimateTau(data=data, ties=ties, from=from, to=to,
                       nmin=nmin, max.length=max.length,
                       cutoff=cutoff, measure=measure,
                       log.like=log.like,
                       calculate.tau=F)$tau.max
#we calculate the criticalCutoff-attribute for each node of
#each tau.max at t
for (time in from:to) {
  tau.max[[time]]$Do(function(node) {node$criticalCutoff <-
    calculatePruningMeasure(node, measure)})
}
#we aggregate the criticalCutoff-attribute with regards to the
#time point sets and weights
for (time in from:to) {
  #fetch the time points considered
  times <- time.sets[[time]]
  #fetch the corresponding weights
  weights <- weight.sets[[time]]
  #renew the criticalCutoff of every node by aggregating the
  #weighed criticalCutoffs of the node at all time points
  tau.max[[time]]$Do(function(node) {
```

```
    D <- 0
    for (i in 1:length(times)) {
      t <- times[i]
      w <- weights[i]
      d <- Navigate(tau.max[[t]],
             path=node$path[2:length(node$path)])$criticalCutoff
      d <- ifelse(is.null(d), 0 , d)
      D <- D + d*w
    }
    node$aggregatedCriticalCutoff <- D
  })
}
for (time in from:to) {
  tau.max[[time]]$Do(function(node) {
    node$criticalCutoff <- node$aggregatedCriticalCutoff
    node$RemoveAttribute("aggregatedCriticalCutoff")
  })
}
#obtain the estimator by pruning tau.max using the aggregated
#critical cutoffs
K <- cutoff*sqrt(nobs*log(nobs))
for (time in from:to) {
  tau[[time]] <- Clone(tau.max[[time]])
  Prune(tau[[time]], function(child) child$criticalCutoff>=K)
}
#if log.like is true, we perform the calculation of the
#loglikelihood function
log.like <- ifelse(isTRUE(log.like),
                   calculateLogLikelihood(data=data, ties=ties,
                                          tau=tau), NA)
#we calculate the model complexity of the fitted tiVLMC
model.complexity <- calculateModelComplexity(alphabet=alphabet,
                                             tau=tau)
#we save the estimated tau, the tau.max, the call, the
#model complexity and the value of the loglikelihood function
estimate <- list(call=match.call(), tau=tau, tau.max=tau.max,
                 model.complexity=model.complexity,
```

```
                    log.like=log.like)
  return(estimate)
}
```

R-Code 11.1: The estimateMovingAverageTau-function

In the formulation of Theorem 11.10 and Theorem 11.11, each vector $(g_r)_{r\in\mathscr{T}_t}$ of weights at a time $1 \leq t \leq T-1$ has to be a deterministic element of $\mathbb{R}_{>0}^{|\mathscr{T}_t|}$. But, if certain conditions are met, we cannot only allow the weights to be random variables but they can even be allowed to vary with the different nodes that are under consideration for pruning (and not only vary with the time t). E.g. this allows for weights that, at every time point t, take the observation size $\tilde{N}_{t+}(wu)$ into respect, when evaluating whether the leaf wu, $u \in M$, should be pruned down to w or whether it should not. We state and prove this within the scope of the following Theorem 11.12.

Theorem 11.12. *If one sets*

$$g_r := \frac{\sqrt{n \log n}}{\sqrt{\tilde{N}_{r+}(wu) \log \tilde{N}_{r+}(wu)}}$$

as the input of Algorithm 11.8, Theorem 11.10 and Theorem 11.11 remain true.

Proof of Theorem 11.12. We start by proving that underestimation does not occur, i.e. by confirming Theorem 11.10. Fix t. Let $wu \in \tau_t^{\mathsf{t}}$, $u \in M$, be a leaf of the true context tree. With the same arguments as in the proof of Theorem 11.10 we get the existence of a threshold integer $n_0 \in \mathbb{N}$ such that

$$\tilde{\Delta}_t(wu) \geq C\sqrt{n \log n}$$

holds for all $n \geq n_0$. Since naturally

$$\tilde{N}_{t+}(wu) \leq n$$

and therefore

$$C\sqrt{\tilde{N}_{t+}(wu) \log \tilde{N}_{t+}(wu)} \leq C\sqrt{n \log n},$$

it immediately follows that

$$g_t\tilde{\Delta}_t(wu) \geq C\sqrt{n\log n}$$

for an even smaller threshold integer $n_1 \in \mathbb{N}_0$. The claim that we do not underestimate now follows directly by reusing the remaining arguments of the original proof of Theorem 11.10.
To finish the proof of this theorem we need to show that Theorem 11.11 stays true.
"$\Rightarrow$":
Since $\tilde{N}_{t_0+}(wu) \leq n$, we have

$$C\sqrt{\tilde{N}_{t_0+}(wu)\log\tilde{N}_{t_0+}(wu)} \leq C\sqrt{n\log n}.$$

Thus it follows that if

$$\tilde{\Delta}_{t_0}(wu) \geq C\sqrt{n\log n}$$

holds,

$$\tilde{\Delta}_{t_0}(wu) \geq C\sqrt{\tilde{N}_{t_0+}(wu)\log\tilde{N}_{t_0+}(wu)} \tag{11.5}$$

is also true and, by rearranging, hence

$$g_{t_0}\tilde{\Delta}_{t_0}(wu) \geq C\sqrt{n\log n}.$$

Thus it follows analogously to the proof of Theorem 11.11 that we will eventually overestimate as $n \to \infty$, i.e. that

$$wu \in \dot{\tau}_t^{\mathsf{t}}.$$

"$\Leftarrow$":
Fix t. Let $w \in \tau_t^{\mathsf{t}}$ be a leaf of τ_t and $u \in M$. As argumented in the proof of Theorem 5.8, we can write

$$\tilde{N}_{r+}(wu) = nP_r(wu)j_r(w) + r$$

with $r = \mathcal{O}(\sqrt{n\log\log n})$. Therefore

$$\begin{aligned}
&\sqrt{\tilde{N}_{r+}(wu)\log\tilde{N}_{r+}(wu)}\\
=&\sqrt{(nP_r(wu)j_r(w)+r)\log\left(nP_r(wu)j_r(w)+r\right)}\\
=&\sqrt{nP_r(wu)j_r(w)\log\left(nP_r(wu)j_r(w)+r\right)+r\log\left(nP_r(wu)j_r(w)+r\right)}\\
=&\mathcal{O}\left(n\log n\right).
\end{aligned}$$

Since

$$g_r\tilde{\Delta}_r(wu) < C\sqrt{n\log n}$$

is equivalent to

$$\tilde{\Delta}_r(wu) < C\sqrt{\tilde{N}_{r+}(wu)\log\tilde{N}_{r+}(wu)}, \tag{11.6}$$

it follows analogously to the proof of Theorem 11.11 that the probability that $wu \in \dot{\tau}_t^{\mathsf{t}}$ vanishes as $n \to \infty$. □

The rationale behind the weights specified in above Theorem 11.12 is as follows: Assume wu and vx with $u, x \in M$ are two leaves that are under consideration for pruning at the time r. Further assume that both, wu as well as vx, carry the same amount of information when measured with the censored time-sensitive pruning measure, i.e.

$$\tilde{\Delta}_r(wu) = \tilde{\Delta}_r(vw).$$

At the same time one leaf, w.l.o.g. let it be wu, is observed more frequently than the other in the sense that

$$\tilde{N}_{r+}(wu) > \tilde{N}_{r+}(vx) \tag{11.7}$$

holds. This difference is not reflected in the pruning decision, since for both leaves we base our pruning decision on exactly the same inequality

$$\tilde{\Delta}_r(wu) = \tilde{\Delta}_r(vx) < C\sqrt{n\log\log n}.$$

By using the weights defined in Theorem 11.12 we now base our pruning decision on whether

$$\tilde{\Delta}_r(wu) < C\sqrt{\tilde{N}_{r+}(wu)\log\tilde{N}_{r+}(wu)}$$

and

$$\tilde{\Delta}_r(vx) < C\sqrt{\tilde{N}_{r+}(vx)\log\tilde{N}_{r+}(vx)}$$

are true or not for wu and vx respectively, cf. the two equations (11.5) and (11.6). Now the different frequency counts are taken into consideration: Since (11.7)

$$C\sqrt{\tilde{N}_{r+}(vx)\log\tilde{N}_{r+}(vx)} < C\sqrt{\tilde{N}_{r+}(wu)\log\tilde{N}_{r+}(wu)}$$

also holds and thus we require less information from the less observed leaf vx compared to wu in order to keep it in our estimate. Since wu was observed more frequently, we expect our measurement of information to be more significant/robust: The probability of underestimating the information gained by keeping wu instead of only w is expected to be decreasing. This allows us to tighten up the threshold in our pruning decision.

11.3 LASSO-tiVLMC

Instead of directly including a smoothing step into Algorithm 5.6, as we did in Algorithm 11.8 in Chapter 11.2, one can also perform smoothing as an additional post-processing procedure, e.g. within the tuning process of the algorithm. Transferring the principle idea of the LASSO to the setting of tiVLMC seems like a promising approach. LASSO stands for "Least Angle Shrinkage and Selection Operator" and is a feature selection technique mainly developed by and in Tibshirani (1996): Simply said, the LASSO chooses the most relevant predictors to explain a dependent variable from a set of potential predictors, yielding a model fit. Mathematically, the LASSO chooses the vector $\boldsymbol{\beta}_{\mathsf{LASSO}}$ of model coefficients as the solution of the minimization problem

$$\min_{\boldsymbol{\beta}} \mathrm{MSE}(\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_{L^1},$$

where $\lambda > 0$ is a tuning parameter and MSE denotes the mean squared error. Due to the angularity of L^1-level sets, using the LASSO-penalty

$$\|\boldsymbol{\beta}\|_{L^1}$$

favours model coefficients valued zero, therefore excluding the corresponding predictor from the model fit and hence performing feature selection.
We utilize the idea of the LASSO by transferring it to the concept of break-point-tiVLMC, cf. Chapter 11.1. Instead of letting the LASSO decide on whether to keep or exclude a predictor variable in the model fit, we let it decide at which of all the potential time points the context tree of the tiVLMC-fit is allowed

to change. In this way a break-point-tiVLMC-fit is obtained, but in contrast to the original break-point-tiVLMC-concept, there is no need to pre-determine the number k of allowed context tree changes any more. To do so, we introduce an $(T-1)$-dimensional binary index vector

$$\boldsymbol{b} = (b_1, ..., b_{T-1}).$$

The t-th coordinate

$$b_t := \mathbb{1}(\tilde{\tau}_t \neq \tilde{\tau}_{t+1})$$

indicates whether the estimated context tree changes from t to $t+1$ or whether it does not. Then we tune Algorithm 5.6 by choosing the cutoff C_{LASSO} given by

$$C_{\mathsf{LASSO}} = \underset{C>0}{\arg\min}\ \mathrm{Loss}(C) + f(C)\|\boldsymbol{b}\|_{L^1},$$

thereby balancing the goodness of fit and the movement in the context tree. Small values of the cutoff C yield more complex tiVLMC-fits. Therefore, not necessarily but most likely, the number of tree changes $\|\boldsymbol{b}\|_{L^1}$ increases as C decreases. Thus the function f should be monotonically decreasing in C to penalize an increasing number of tree changes. Possible choices for the loss function Loss are the negative estimated log-likelihood function $-\log L$, cf. (6.2), or the mean squared error MSE.

Note that every possible cutoff value corresponds to one tiVLMC-fit, i.e. the candidate set of models is the same in this LASSO-approach as it is in the original one. We just choose the cutoff differently. Hence models that are not part of this candidate set cannot be the result of this post-processing procedure. This is a difference compared to the moving-average-tiVLMC presented in Chapter 11.2: As a direct consequence of the short remark after Definition 11.7 the candidate set of all moving-average-tiVLMC is (arbitrarily) larger.

The output of this approach is a break-point-tiVLMC-model where the break points are implicitly given by the binary index vector $\boldsymbol{b}_{\mathsf{LASSO}}$ corresponding to the model fitted with the cutoff C_{LASSO}. E.g. the indicator vector

$$\boldsymbol{b}_{\mathsf{LASSO}} = (0, 0, 1, 0, 0, 1)$$

would imply two break points at times 3 and 6, the fitted model therefore having

$$(\tilde{\tau}^{\text{super}}_{1:3}, \tilde{\tau}^{\text{super}}_{1:3}, \tilde{\tau}^{\text{super}}_{1:3}, \tilde{\tau}^{\text{super}}_{4:6}, \tilde{\tau}^{\text{super}}_{4:6}, \tilde{\tau}^{\text{super}}_{4:6}, \tilde{\tau}_7)$$

as its context tree.
Numerically we evaluate $\|\boldsymbol{b}\|_{L^1}$ by calling the treeChanges-function in R, cf. the following R-Code 11.2. The function relies on the identicalNodes-function displayed in R-Code 11.3.

```
treeChanges <- function(trees) {
  #initialize the output
  penalty <- 0
  #iterate through the input list of trees
  #for each successive pair of list elements
  #we check whether they are unequal
  for (t in 1:(length(trees)-1)) {
    if (!identicalNodes(trees[[t]], trees[[t+1]])) {
      #if they are unequal, we increase the output by one
      penalty <- penalty+1
    }
  }
  return(penalty)
}
```

R-Code 11.2: The treeChanges-function

```
identicalNodes <- function(tree.1, tree.2) {
  #test whether the node structure of two input
  #trees is identical or whether it is not
  #to do so, check all node-paths for equality
  tree.1.paths <- sort(ToDataFrameTable(tree.1, "pathString"))
  tree.2.paths <- sort(ToDataFrameTable(tree.2, "pathString"))
  return(identical(tree.1.paths, tree.2.paths))
}
```

R-Code 11.3: The identicalNodes-function

Now, since the described technique just automates the procedure of first choosing the number k of break points and secondly optimizing their position in time via the choice of the cutoff parameter C, Corollary 5.9 directly implies that this

LASSO-alike approach outputs a consistent context tree estimator, that is, it equals the true context tree of the data-generating tiVLMC with probability one as the observation size diverges to infinity.

11.4 Cutoff paths

The second step of Algorithm 5.6 prunes the maximal tree $\tau_{\max}$ consisting of all sequences observed at least $n_{\min}$ times down to the estimated context tree $\tilde{\tau}_t$ at time t. To do so, for each leaf of $\tau_{\max}$ a pruning decision is made based on the censored time-sensitive pruning measure $\tilde{\Delta}_t$, cf. Definition 5.5, and the cutoff value C. This is iteratively repeated until no more leaves can be pruned. The resulting tree is the context tree estimator $\tilde{\tau}_t$ at time t. While the value of the censored time-sensitive pruning measure is data-driven, the cutoff value C is a tuning parameter input by the user. Since the pruning decision is binary and based on a continuous inequality in C, we can always calculate the maximal C such that a leaf is not pruned. Recall that smaller cutoff values yield larger trees, cf. (6.1). A leaf wu, $u \in M$, is pruned down to w if and only if

$$\tilde{\Delta}_t(wu) < C\sqrt{n \log n}, \tag{11.8}$$

thus the maximal cutoff value $C_t^{\text{crit}}(wu)$ such that wu is not pruned down to w is given by

$$C_t^{\text{crit}}(wu) := \frac{\tilde{\Delta}_t(wu)}{\sqrt{n \log n}}. \tag{11.9}$$

Here we assume that there are at least two observations under study, i.e. $n \geq 2$. This assumption implies $\sqrt{n \log n} > 0$ and is necessary here. The assumption is natural anyway, since inference from a single observation is pointless within this framework. If the empirical transition probability measures $\tilde{P}_t(\cdot \mid wu)$ and $\tilde{P}_t(\cdot \mid w)$ are equal, $C_t^{\text{crit}}(wu) = 0$ holds, which translates to pruning wu down to w regardless of the value of the cutoff $C > 0$. This is implied by Definition 5.5 and the fact that the Kullback-Leibler divergence D between two probability measures, cf. Definition 4.1, is zero if and only if the two measures are equal.
Since the right-hand side of equation (11.9) must be evaluated to decide whether (11.8) holds, i.e. whether to prune or not to prune, it comes with no extra computational effort to compute $C_t^{\text{crit}}(wu)$. For every node $wu \in \tau_{\max}$ the maximal cutoff value $C_t^{\text{crit}}(wu)$ is stored in an attribute of `tau.max[[t]]`. E.g. if $w = u = \mathsf{a}$, we

can access $C_t^{\text{crit}}(\mathsf{aa})\sqrt{n \log n}$ via tau.max[[t]]$"a"$"a"$criticalCutoff thus obtaining $C_t^{\text{crit}}(\mathsf{aa})$ by dividing by $\sqrt{n \log n}$.
This gives paths of maximal cutoff values $C_t^{\text{crit}}(wu)$ in t. By looking at such a path the user is able to detect the maximal cutoff value that ensures that a leaf wu is not pruned and instead is kept in the model. Of course there are time points t where wu may not be under consideration for pruning, because e.g. the sequence was not observed $n_{\min}$-times, i.e. $\tilde{N}_{t+}(wu) < n_{\min}$. At those time points the choice of the cutoff has no impact on the inclusion of wu in the model.
A real life application of the maximal cutoff paths is when the practitioner has to guarantee that certain state change sequences are part of the model, i.e. part of the context trees. A reason for this can be for example that those sequences trigger payments between the insured and the insurance company by contract and thus have to be included within the model. If this sequence is denoted by w, one can look at the path

$$\left\{C_t^{\text{crit}}(w) : 1 \leq t \leq T-1\right\}$$

and then tune the model only with cutoff values smaller than

$$\min\left\{C_t^{\text{crit}}(w) : 1 \leq t \leq T-1\right\},$$

i.e. construct the set $\mathscr{C}$, cf. Chapter 6.3, such that

$$\max \mathscr{C} \leq \min\left\{C_t^{\text{crit}}(w) : 1 \leq t \leq T-1\right\}.$$

This described procedure is similar to the paths of coefficients within the context of the LASSO, or more generally the elastic net, cf. Tibshirani (1996) or Friedman et al. (2010). The second reference is for the corresponding R-package called glmnet.

12 The joint life insurance

The canlifins data set "contains information of 14,889 contracts in force with a large Canadian insurer over the period December 29, 1988 through December 31, 1993. These contracts are joint and last-survivor annuities [...].", cf. Dutang and Charpentier (2019, page 26). The exact source of the data set, i.e. the name of the Canadian insurance company, is unknown. canlifins is part of the well-known CASdatasets R-package, a collection of actuarial data sets originally put together for the book "Computational Actuarial Science with R", cf. Charpentier (2014). CASdatasets (and therefore also canlifins) is publicly available for download, it is licensed under a GNU General Public Licence.
For each of the 14,889 contracts five features are documented: The age EntryAgeM of the male part of the insured couple at the start of the contract, the age EntryAgeF of the female part of the insured couple at the start of the contract, the time DeathTimeM of death of the male measured from the start of the observation period and the time DeathTimeF of death of the female measured from the start of the observation period. If the male or the female do not die within the observation period, DeathTimeM or DeathTimeF are assigned the value zero. If an annuity guarantee was part of the contract, the fifth feature AnnuityExpiredM documents the date the guarantee expired.
In the following our main goal is to obtain insights into the dependency structure between the male and the female part of the joint life insurance by fitting tiVLMC-models and applying the developed smoothing techniques of Chapter 11.

12.1 Data preparation

Six observations contained in the data set are displayed in the following R-Code 12.1.

```
##    EntryAgeM EntryAgeF DeathTimeM DeathTimeF AnnuityExpiredM
## 1    60.6749   62.1217     0.0000     0.0000          5.0055
## 2    69.1463   68.4249     0.0000     0.0000          5.0055
## 3    66.1612   64.9973     0.0000     0.0000          1.6655
## 4    58.0751   60.0039     0.0000     0.0000          1.6655
## 14   77.0575   75.3744     0.0000     2.5450          5.0055
## 16   92.9550   91.6600     2.5696     1.2144          5.0055
```

R-Code 12.1: Six out of 14,889 observations of the canlifins data set

E.g. the male part of the first contract is 60.67 years old when the contract was closed and the corresponding female part is 62.12 years old at that time (both numbers rounded to two decimal digits). Neither the male nor the female die within the observation period, therefore DeathTimeM=DeathTimeF=0.
First we reformat the data set: Since at this point we are not interested in annuities, we drop AnnuityExpiredM from the data. Next we introduce ExitAgeM and ExitAgeF, the ages of the (possibly dead) male and female at the end of the observation period. Lastly we replace DeathTimeM and DeathTimeF with the actual ages at death, DeathAgeM and DeathAgeF. If the observation period is survived by the male or female, we assign DeathAgeM=NA or DeathAgeF=NA respectively. After the reformatting the observations displayed in R-Code 12.1 look as follows:

```
##    EntryAgeM EntryAgeF DeathAgeM DeathAgeF ExitAgeM ExitAgeF
## 1    60.6749   62.1217        NA        NA  65.6804  67.1272
## 2    69.1463   68.4249        NA        NA  74.1518  73.4304
## 3    66.1612   64.9973        NA        NA  71.1667  70.0028
## 4    58.0751   60.0039        NA        NA  63.0806  65.0094
## 14   77.0575   75.3744        NA   77.9194  82.0630  80.3799
## 16   92.9550   91.6600   95.5246   92.8744  97.9605  96.6655
```

R-Code 12.2: Six out of 14,889 reformatted observations of the canlifins data set

Since the data is documented in a continuous fashion, we have to discretize it in order to apply the tiVLMC-techniques that require discrete data as input. We note that in practice applying methods designed to handle continuous data should most likely be favoured here, but the discretization is done in order to obtain a second, publicly available and more computational-friendly (compared

to the more complex SIGNAL IDUNA data set data.signal used in Chapter 10) data set to experiment with the developed tiVLMC-techniques.
We work on a state space M consisting of four states: aa, ad, da and dd. a indicates that the insured is alive and d indicates that the insured is dead. The first character corresponds to the male part of the joint life insurance, the second character to the female part. E.g. da indicates a dead male and an alive female.
To generate an easy to handle, slim data set, we choose a discretization period of one year. Here we exemplarily generate the data set data.female, where t denotes the age of the female and therefore we take the female perspective.
We describe the advanced data preparation. Since not all females close (and therefore exit) the contract at the same age, the data is subject to left and right censoring. An observation is censored at age t if $t \notin [\text{EntryAgeF}, \text{ExitAgeF})$ holds for this observation. If it is not censored at t, it is assigned the state aa if the female is alive at age t and her male partner (which in general is not of age t at this time) is alive. We assign da if the female is alive at age t and her male partner is dead and ad if it is the other way round. Lastly the state dd is assigned at t if both partners are dead when the female is aged t.
The whole data preparation is displayed in R-Code 12.3.

```
#first we add a column with the age of death
#if the individual did not die within the observation period we
#fill in NA
canlifins[canlifins[, "DeathTimeM"]==0, "DeathTimeM"] <- NA
canlifins[, 6] <- canlifins[, "EntryAgeM"]+
                  canlifins[, "DeathTimeM"]
colnames(canlifins)[6] <- "DeathAgeM"
canlifins[canlifins[, "DeathTimeF"]==0, "DeathTimeF"] <- NA
canlifins[, 7] <- canlifins[, "EntryAgeF"]+
                  canlifins[, "DeathTimeF"]
colnames(canlifins)[7] <- "DeathAgeF"
#second we add a column containing the age at observation end
canlifins[, 8] <- canlifins[, "EntryAgeM"]+5.0055
colnames(canlifins)[8] <- "ExitAgeM"
canlifins[, 9] <- canlifins[, "EntryAgeF"]+5.0055
colnames(canlifins)[9] <- "ExitAgeF"
#third we drop the non-relevant columns from the data set
canlifins <- canlifins[, c("EntryAgeM", "EntryAgeF", "DeathAgeM",
```

```
                              "DeathAgeF", "ExitAgeM", "ExitAgeF")]
#we reformat the data
#first the dimensions of the new data set are calculated
#the maximal exit age in years
max.age <- ceiling(max(canlifins[, c("ExitAgeM", "ExitAgeF")]))
#the number of observations
nobs <- dim(canlifins)[1]
#we now construct the dataset data.female
#in data.female t denotes the age of the female
#we initialize data.female with the correct dimensions
data.female <- matrix(rep(0, times=max.age*nobs), ncol=max.age)
colnames(data.female) <- 1:max.age
#we iterate through each observation and each age t and
#assign the correct state
for (obs in 1:nobs) {
  #grab the observation
  observation <- canlifins[obs, ]
  for (t in 1:max.age) {
    #check whether the observation is censored
    entered <- (observation["EntryAgeF"] <= t) &&
               (t < observation["ExitAgeF"])
    #if it is not censored, assign the correct state
    if (entered) {
      age.female <- t
      age.difference <- observation["EntryAgeF"] -
                        observation["EntryAgeM"]
      age.male <- age.female-age.difference
      death.female <- observation["DeathAgeF"]
      death.male <- observation["DeathAgeM"]
      if (is.na(death.female)) {
        status.female <- "a"
      } else if (age.female < death.female) {
        status.female <- "a"
      } else {
        status.female <- "d"
      }
      if (is.na(death.male)) {
```

```
        status.male <- "a"
      } else if (age.male < death.male) {
        status.male <- "a"
      } else {
        status.male <- "d"
      }
      state <- paste(status.male, status.female, sep="")
    #if it is censored, indicate so by assigning NA
    } else {
      state <- NA
    }
    data.female[obs, t] <- state
  }
}
#we rename the rows
rownames(data.female) <- 1:dim(data.female)[1]
```

R-Code 12.3: The data preparation steps performed on the canlifins data set

After the discretization the 14th observation, which was already printed in the Codes 12.1 and 12.2, looks as displayed in R-Code 12.4. At the ages t where the status is not printed the observation is censored.

```
print(data.female[14, 75:81])

##   75   76   77   78   79   80   81
##   NA "aa" "aa" "ad" "ad" "ad"   NA
```

R-Code 12.4: The 14-th observation of data.female after the discretization

12.2 The LASSO-tiVLMC-model

Now we fit a LASSO-tiVLMC-model, cf. Chapter 11.3, to data.female. To measure model precision we choose the negative estimated log-likelihood function, i.e.

$$\text{Loss}(C) = -\log L\,(C),$$

cf. Chapter 6.1, and to measure model complexity we choose

$$f : \mathbb{R}_{>0} \to \mathbb{R}_{>0},\ C \mapsto \frac{1}{C}.$$

We tune the LASSO-fit on the cutoff grid

$$\begin{aligned}\mathscr{C} = \{&0.0000001, 0.0000002, ..., 0.0000099,\\ &0.00001, 0.00002, ..., 0.00099,\\ &0.001, 0.002, ..., 0.099\\ &0.1, 0.2, ..., 1\}\end{aligned}$$

that consists of 307 cutoff values that are sampled more and more finely as they approach zero. For comparison we also fit the AIC- and the BIC-tuned tiVLMC-models. Tuning via AIC leads to a cutoff of $C_{\text{AIC}} = 0.016$, using BIC we get $C_{\text{BIC}} = 0.034$ and the LASSO-tuning gives the largest cutoff of $C_{\text{LASSO}} = 0.075$ and therefore the smallest model. While in total 14 different trees appear as estimated context trees within the three model fits, the AIC-tuned model contributes 11, the BIC-tuned model nine and the LASSO-tuned model seven. The LASSO-model changes its context tree only 10 times, while the AIC-model, as well as the BIC-model, change their context trees 15 times.
The following Figure 12.1 displays the movement of the context trees as time passes by.

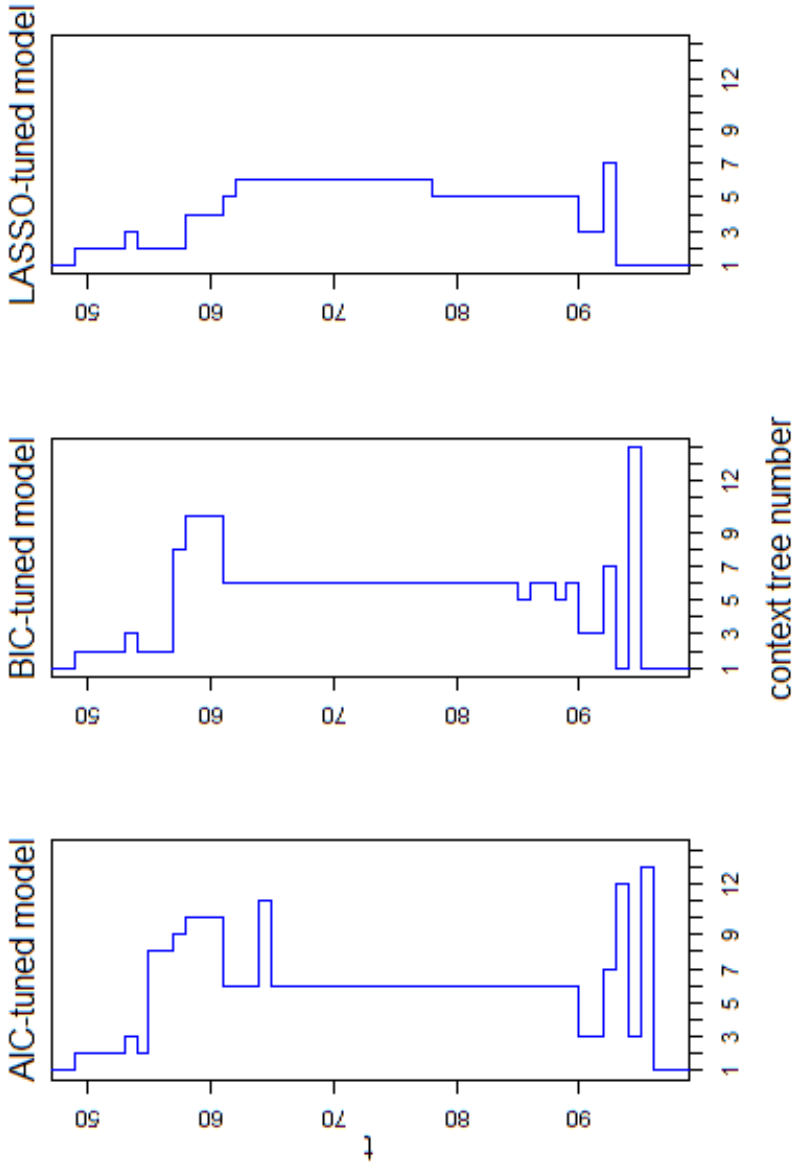

Figure 12.1: The movement of the differently estimated context trees of the AIC-tuned tiVLMC-model (left), the BIC-tuned tiVLMC-model (middle) and LASSO-tuned tiVLMC-model (right) between all occurring trees as the time t passes

As the numbers already suggest, one can clearly see the smoothing effect in above figure. Thus, if one chooses f and Loss in a way that is reasonable, a LASSO-tiVLMC-model that is easier to interpret and communicate when compared to the classical tiVLMC-model can be obtained.
Since the focus of this chapter is on the smoothing effect, we skip plotting all 14 occurring context tree estimates at this point. But, to ensure the overall conclusiveness, the trees are printed in the Appendix C, cf. Figures C.1-C.14.

12.3 The moving-average-tiVLMC-models

As an alternative to the LASSO-tiVLMC proposed in the previous Chapter 12.2 we now consider different moving-average-tiVLMC on data.female. In order to simplify the original AIC-tuned tiVLMC-model, we smoothen the context tree movement by using different smoothing kernels, i.e. using different combinations of time point sets and weight vectors as input in Algorithm 11.8. Then the cutoff is chosen in order to minimize AIC, i.e. $C = C_{\text{AIC}}$.
We consider the following combinations of time point sets and weight vectors:

1. A "flat line" kernel always weighing with the same weight of one at all time points , i.e. $\mathscr{T}_t = \{1, ..., T-1\}$ and $g_t = 1$ for all $t = 1, ..., T-1$.

2. A "pyramid" kernel that puts the highest weight at the current time point and decreasing weights at the closest two time points in the past and the future, i.e. $\mathscr{T}_t = \{t-2, t-1, t, t+1, t+2\}$ and weights $(0.55, 1.11, 1.67, 1.11, 0.55)$ for $3 \leq t \leq T-3$.

3. A "downwards stairs" kernel that puts the highest weight at the current time point and decreasing weights at the closest two time points in the future, i.e. $\mathscr{T}_t = \{t, t+1, t+2\}$ and weights $(1.5, 1, 0.5)$ for $1 \leq t \leq T-3$.

4. An "upwards stairs" kernel that puts the highest weight at the current time point and decreasing weights at the closest two time points in the past, i.e. $\mathscr{T}_t = \{t-2, t-1, t\}$ and weights $(0.5, 1, 1.5)$ for $3 \leq t \leq T-1$.

In all cases the kernels fade in at the left margin and fade out at the right margin, i.e. they are always centred at the present time point t. This is usual practice for kernel smoothing techniques, e.g. cf. Ghosh (2018, page 25). We have not explicitly stated this in above listing. In case 2 e.g. this translates to $\mathscr{T}_1 = \{1, 2, 3\}$ with weights $(1.67, 1.11, 0.55)$, $\mathscr{T}_2 = \{1, 2, 3, 4\}$ with weights $(1.11, 1.67, 1.11, 0.55)$

at the left margin. On the right side we have $\mathscr{T}_{T-2} = \{T-4, T-3, T-2, T-1\}$ with weights $(0.55, 1.11, 1.67, 1.11)$ and $\mathscr{T}_{T-1} = \{T-3, T-2, T-1\}$ with weights $(0.55, 1.11, 1.67)$.

As already mentioned in the previous Chapter 12.2, the AIC-tuned, original model changes its context tree 15 times. For the "flat line"-moving-average-tiVLMC described in case 1 the number of context tree changes merely reduces to 14. The "pyramid" kernel, case 2, leads to a model fit that alters its context tree 11 times. For the kernel shaped like an "upwards stairs", i.e. case 3, this number is 13 and for the "downwards stairs" it is 10.

Figure 12.2 shows the context tree movements of the differently smoothened models. R-Code 12.5 exemplarily shows how to calculate the moving-average-tiVLMC using the "pyramid" kernel, i.e. case 2, in R using the estimateMovingAverageTau-function.

Note that we considered four combinations of time sets and weight vectors here. Obviously there are infinitely many more one could try. It is also not required that the kernel keeps its shape over time: Theorem 11.10 remains true even if the kernel morphs its shape as time passes by.

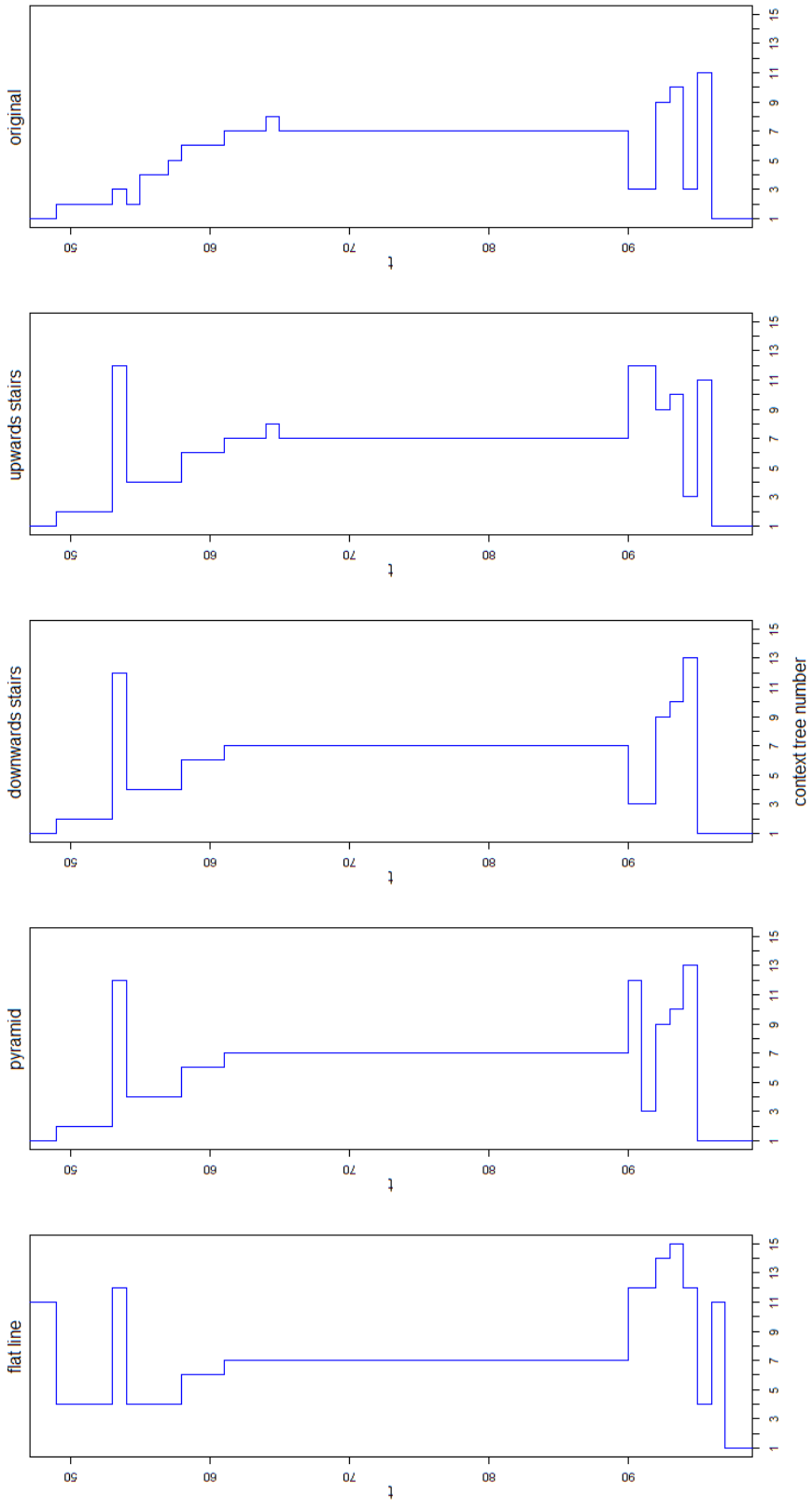

Figure 12.2: The movement of the estimated context tree of the moving-average-tiVLMC-models smoothed by the kernels 1-4 between all occurring trees as the time t passes

```
#initialize the list of weight vectors
ws <- list()
#define the weight vectors for every time t
ws[[1]] <- c(1.67, 1.11, 0.55)
ws[[2]] <- c(1.11, 1.67, 1.11, 0.55)
for (t in 3:(dim(data.female)[2]-3)) {
  ws[[t]] <- c(0.55, 1.11, 1.67, 1.11, 0.55)
}
ws[[dim(data.female)[2]-2]] <- c(0.55, 1.11, 1.67, 1.11)
ws[[dim(data.female)[2]-1]] <- c(0.55, 1.11, 1.67)
#initialize the list of time point sets
ts <- list()
#define the time point set for each time t
ts[[1]] <- c(1, 2, 3)
ts[[2]] <- c(1, 2, 3, 4)
for (t in 3:(dim(data.female)[2]-3)) {
ts[[t]] <- c(t-2, t-1, t, t+1, t+2)
}
ts[[dim(data.female)[2]-2]] <- c(dim(data.female)[2]-4,
                                 dim(data.female)[2]-3,
                                 dim(data.female)[2]-2,
                                 dim(data.female)[2]-1)
ts[[dim(data.female)[2]-1]] <- c(dim(data.female)[2]-3,
                                 dim(data.female)[2]-2,
                                 dim(data.female)[2]-1)
#fit the moving-average-tiVLMC-model
tau <- estimateMovingAverageTau(data=data.female, time.sets=ts,
                                weight.sets=ws, cutoff=0.061,
                                measure="KLD")
```

R-Code 12.5: Fitting the moving-average-tiVLMC on the data.female data set using the "pyramid" kernel

The cutoff used in above R-Code 12.5 is the result of a preceding tuning performed to minimize AIC. Here we do not display this procedure since it is completely analogous to the two examples in Chapter 8.2 (or R-Code 8.12).

13 Reflection and directions for future research

In this thesis we have developed the model class of tiVLMC, cf. Chapter 3. We have discussed how tiVLMC meet the special requirements of insurance modelling listed in Chapter 1.3. tiVLMC are e.g. capable of displaying time-inhomogeneous dependencies in data. To be of practical use, a fitting algorithm for tiVLMC-models has been established and proven to be consistent, cf. Chapter 4, even in the presence of censoring, cf. Chapter 5. We have extended known model selection techniques to the tiVLMC-setup, cf. Chapter 6, we have evaluated different smoothing approaches for tiVLMC-fits, cf. Chapter 11. We have discussed how an insurance contract can be priced by using a tiVLMC-model, cf. Chapter 9. The fitting procedure, the tuning procedure and the pricing procedure as well have all been implemented in R, cf. Chapter 7, and the implementations have been documented. We have applied the techniques on differently generated data sets, cf. Chapter 8, and real life data sets, cf. Chapters 10.3 and 12 and we have discussed the results.
Summarizing we can say that this thesis offers everything a potential user of tiVLMC-models needs, if her or his working environment fits our theoretical framework. The various examples can be adapted and/or generalized to the reader's own data setup and are not limited to insurance applications.
Nevertheless there is always room for further research. Here we list possible advancements by chapters:

Chapter 2:
As our theoretical framework we chose an environment where the data consists of time-discrete, independent and identically distributed observations.
A follow-up question to this thesis is whether the developed tiVLMC-models can be generalized in a way so that they also work in a time-continuous framework. Such a generalization would make discretizing continuous data (as we did e.g. in Chapter 12.1) unnecessary. Time-continuous tiVLMC would include semi-Markov processes.
One could also investigate whether the obtained results, or at least part of them, will still hold if we allow the observations to be dependent on each other. Since tools like the strong law of large numbers A.1 cannot be applied in that case, one would have to find different proving techniques. If observations were allowed to depend on each other, we could account for more behavioural effects like the following example: Two insured living in the same household are more likely to lapse their insurance contract at the same time compared to two insured living in

separate households. Assuming that the observations are independent we cannot model such traits.

Chapter 5:
If we stick with the time-discrete, independent and identically distributed observations, one could investigate e.g. whether the censoring assumptions (H.1)-(H.5) could be partly eliminated or replaced by weaker versions. Using statistical wording, it would be desirable to be able to extend the theoretical results from the current "missing completely at random" environment to a "missing at random" one.

Chapter 6:
When we discussed model selection we focused on information criteria, cf. Chapter 6.3. One could also introduce different tuning techniques. E.g. a cross-validation procedure for choosing the cutoff could be implemented.

Chapter 7:
The implementation estimateTau of Algorithm 5.9 supports the Kullback-Leibler divergence and the L^1-norm as possible inputs for measure. However, as we have learned in Chapter 5.3, other norms also lead to consistent context tree estimates. Thus one could generalize estimateTau to allow arbitrary norms as input of measure.
To maximize the usability of the developed techniques the computational effort needed to run the implementations should be as small as possible. The user's experience handling the R-implementations should also be as good as possible. However the focus of this thesis is on the mathematical theory. It would be beneficial if someone proficient in coding re-evaluated the most efficient options for implementing the algorithms. As it is good practice, the various R-scripts could also be combined into one package.
Lastly, one could consider e.g. using the shiny-package, cf. Chang et al. (2020), in order to develop a graphical user interface which would improve the usability of the developed techniques for the majority of practitioners.

Chapter 9:
We derived a calculation formula for the prospective reserve in a tiVLMC-model, cf. Definition 9.10. Evaluating this formula for coverages with long contract durations is numerically infeasible: The outer sum runs over $|M^{t_e-t}| = m^{t_e-t}$ sum-

mands. If the state space contains e.g. $m = 6$ states and the contract runs from $t_s = 18$ to $t_e = 67$,

$$6^{67-18} = 1.34713546244127 \cdot 10^{38}$$

summands would have to be calculated. Even with access to High-Performance Computing facilities finishing such calculations in an acceptable amount of time is impossible. Here more research is required to derive a computation formula that can be evaluated faster. In the classical setup the Markov assumption prevents this problem. In a similar manner one should evaluate ways of including the dependency structure of the tiVLMC even more into the computation formula. Additionally it is also an important question whether a recursive Thiele-alike computation formula can be derived.

Since we obtain consistent context tree estimates for any norm, the follow-up question arises which norm is suited best to achieve precise approximations of the prospective reserves.

References

Akaike, H.
1998. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, Pp. 199–213. Springer.

Analytics, R. and S. Weston
2015. foreach: Provides foreach looping construct for R. *R package version 1.5.0*, 1(3):1.

Anderson, T. W. and L. A. Goodman
1957. Statistical inference about Markov chains. *The Annals of Mathematical Statistics*, Pp. 89–110.

Bachmann, P.
1904. Analytische Zahlentheorie. In *Encyklopädie der Mathematischen Wissenschaften mit Einschluss ihrer Anwendungen*, Pp. 636–674. Springer.

Begleiter, R., R. El-Yaniv, and G. Yona
2004. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, 22:385–421.

Bejerano, G. and G. Yona
2001. Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. *Bioinformatics*, 17(1):23–43.

Bellman, R.
1966. Dynamic programming. *Science*, 153(3731):34–37.

Bercu, B., B. Delyon, and E. Rio
2015. *Concentration inequalities for sums and martingales*. Springer.

Billingsley, P.
1995. *Probability and Measure*, Wiley Series in Probability and Statistics. Wiley.

Böhme, R., G. Schwartz, et al.
2010. Modeling Cyber-Insurance: Towards a Unifying Framework. In *WEIS*.

Boltzmann, L.
1884. Über die Eigenschaften monozyklischer und anderer damit verwandter Systeme. *FP Hasenhörl*, 3.

Borges, J. and M. Levene
2007. Evaluating variable-length Markov chain models for analysis of user web navigation sessions. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):441–452.

Bowers, N., H. Gerber, J. Hickman, D. Jones, and C. Nesbitt
1997. *Actuarial Mathematics*. Society of Actuaries.

Breiman, L., J. Friedman, C. Stone, and R. Olshen
1984. *Classification and Regression Trees*, The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis.

Bühlmann, P., A. J. Wyner, et al.
1999. Variable length Markov chains. *The Annals of Statistics*, 27(2):480–513.

Bundesamt für Justiz
2017. Sozialgesetzbuch (SGB) - Elftes Buch (XI) - § 15 Ermittlung des Grades der Pflegebedürftigkeit. `https://www.gesetze-im-internet.de/sgb_11/__15.html`, Webpage accessed: 2020-07-17.

Bundesamt für Justiz
2020. Sozialgesetzbuch (SGB) - Elftes Buch (XI) - § 14 Begriff der Pflegebedürftigkeit. `https://www.gesetze-im-internet.de/sgb_11/__14.html`, Webpage accessed: 2020-07-17.

Bundestag
2012. Gesetz zur Neuausrichtung der Pflegeversicherung (Pflege-Neuausrichtungs-Gesetz - PNG). `http://dip21.bundestag.de/dip21/btd/17/093/1709369.pdf`, Webpage accessed: 2020-07-15.

Bundestag
2015. Zweites Gesetz zur Stärkung der pflegerischen Versorgung und zur Änderung weiterer Vorschriften (Zweites Pflegestärkungsgesetz - PSGII). `https://www.bgbl.de/xaver/bgbl/start.xav?startbk=Bundesanzeiger_BGBl&jumpTo=bgbl115s2424.pdf`, Webpage accessed: 2020-07-15.

Busch, J. R., P. A. Ferrari, et al.
2009. Testing statistical hypothesis on random trees and applications to the protein classification problem. *The Annals of Applied Statistics*, 3(2):542–563.

Chang, W., J. Cheng, J. Allaire, Y. Xie, and J. McPherson
2020. shiny: Web Application Framework for R. R package version 1.5.0.

Charpentier, A.
2014. *Computational actuarial science with R.* CRC Press.

Christiansen, M. C. and C. Furrer
2020. Dynamics of state-wise prospective reserves in the presence of non-monotone information. *arXiv preprint arXiv:2003.02173*.

Claeskens, G., N. L. Hjort, et al.
2008. Model selection and model averaging. *Cambridge Books*.

Consonni, G. and P. Veronese
1995. A Bayesian method for combining results from several binomial experiments. *Journal of the American Statistical Association*, 90(431):935–944.

Cover, T. M. and J. A. Thomas
1991. Elements of information theory. *Wiley Series in Telecommunications*.

Cox, D. R. and H. D. Miller
1977. *The theory of stochastic processes*, volume 134. CRC Press.

Csiszár, I. and Z. Talata
2006. Context tree estimation for not necessarily finite memory processes, via BIC and MDL. *IEEE Transactions on Information Theory*, 52(3):1007–1016.

De Jong, P., G. Z. Heller, et al.
2008. Generalized linear models for insurance data. *Cambridge Books*.

Dragomir, S. S., M. Scholz, and J. Sunde
2000. Some upper bounds for relative entropy and applications. *Computers & Mathematics with Applications*, 39(9-10):91–100.

Duarte, D., A. Galves, and N. L. Garcia
2006. Markov approximation and consistent estimation of unbounded probabilistic suffix trees. *Bulletin of the Brazilian Mathematical Society*, 37(4):581–592.

Dubnov, S., G. Assayag, O. Lartillot, and G. Bejerano
2003. Using machine-learning methods for musical style modeling. *Computer*, 36(10):73–80.

Dutang, C. and A. Charpentier
2019. CASdatasets: Insurance datasets. R package version 1.0-10.

Embrechts, P. et al.
1996. *Actuarial versus financial pricing of insurance*. PhD thesis.

Etemadi, N.
1981. An elementary proof of the strong law of large numbers. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 55(1):119–122.

European Court of Justice
2004. COUNCIL DIRECTIVE 2004/113/EC of 13 December 2004. `https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32004L0113:EN:HTML`, Webpage accessed: 2020-07-17.

European Parliament
2016. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL. `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679`, Webpage accessed: 2020-06-19.

Ferrari, F. and A. Wyner
2003. Estimation of general stationary processes by variable length Markov chains. *Scandinavian Journal of Statistics*, 30(3):459–480.

Frangos, N. E. and S. D. Vrontos
2001. Design of optimal bonus-malus systems with a frequency and a severity component on an individual basis in automobile insurance. *ASTIN Bulletin: The Journal of the IAA*, 31(1):1–22.

Friedman, J., T. Hastie, and R. Tibshirani
2010. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1.

Frigessi, A. and B. Heidergott
2011. Markov Chains. In *International Encyclopedia of Statistical Science*, Pp. 772–775. Springer Berlin Heidelberg.

Gagniuc, P. A.
2017. *Markov chains: from theory to implementation and experimentation.* John Wiley & Sons.

Galves, A., C. Galves, J. E. Garcia, N. L. Garcia, F. Leonardi, et al.
2012. Context tree selection and linguistic rhythm retrieval from written texts. *The Annals of Applied Statistics*, 6(1):186–209.

Galves, A., V. Maume-Deschamps, and B. Schmitt
2008. Exponential inequalities for VLMC empirical trees. *ESAIM: Probability and Statistics*, 12:219–229.

García, J. E., R. Gholizadeh, and V. A. González-López
2017. Linguistic compositions highly volatile in Portuguese. *Cadernos de Estudos Lingüísticos*, 59(3):617–630.

Garivier, A. and F. Leonardi
2011. Context tree selection: A unifying view. *Stochastic Processes and their Applications*, 121(11):2488–2506.

Garrido, J., C. Genest, and J. Schulz
2016. Generalized linear models for dependent frequency and severity of insurance claims. *Insurance: Mathematics and Economics*, 70:205–215.

Ghosh, S.
2018. *Kernel smoothing: Principles, methods and applications.* John Wiley & Sons.

Glur, C.
2018. data.tree: General Purpose Hierarchical Data Structure. R package version 0.7.8.

Gopalakrishnan, T., P. Sengottuvelan, A. Bharathi, and R. Lokeshkumar
2018. An approach to webpage prediction method using variable order Markov model in recommendation systems. *Journal of Internet Technology*, 19(2):415–424.

Gram, J.
1910. Professor Thiele som aktuar. *Dansk Forsikrings Arbog*, 1910:26–37.

Haberman, S. and E. Pitacco
1998. *Actuarial models for disability insurance*. CRC Press.

Halmos, P. R.
2017. *Naive set theory*. Courier Dover Publications.

Harfst, S.
2020. HPC Facilities of the University of Oldenburg. `https://uol.de/fk5/wr/hochleistungsrechnen/hpc-facilities/carl/`, Webpage accessed: 2020-07-17.

Heilmann, W.-R. and K. J. Schröter
2013. *Grundbegriffe der Risikotheorie*. VVW GmbH.

Heitjan, D. F. and S. Basu
1996. Distinguishing "missing at random" and "missing completely at random". *The American Statistician*, 50(3):207–213.

Herath, H. and T. Herath
2011. Copula-based actuarial model for pricing cyber-insurance policies. *Insurance Markets and Companies: Analyses and Actuarial Computations*, 2(1):7–20.

Hess, E. C.
2015. Auswertung eines Datensatzes einer privaten Krankenversicherung.

Hoeffding, W.
1963. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30.

Hoem, J. M.
1969. Markov chain models in life insurance. *Blätter der DGVFM*, 9(2):91–107.

Izrailev, S.
2014. tictoc: Functions for timing R scripts, as well as implementations of Stack and List structures. R package version 1.0.

Jones, L.
1987. *The Collected Works of John W. Tukey: Philosophy and Principles of Data Analysis 1965-1986*, volume 4. CRC Press.

Kendall, M. G. et al.
1948. The advanced theory of statistics. Vols. 1. *The Advanced Theory of Statistics. Vols. 1.*, 1(Ed. 4).

Khintchine, A.
1924. Über einen Satz der Wahrscheinlichkeitsrechnung. *Fundamenta Mathematicae*, 6(1):9–20.

Koller, M.
2012. *Stochastic models in life insurance*. Springer Science & Business Media.

Kolmogorov, A.
1931. Über die analytischen Methoden in der Wahrscheinlichkeitsrechnung. *Mathematische Annalen*, 104(1):415–458.

Kullback, S. and R. A. Leibler
1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Kwon, H.-S. and B. L. Jones
2006. The impact of the determinants of mortality on life insurance and annuities. *Insurance: Mathematics and Economics*, 38(2):271–288.

Larson, S. C.
1931. The shrinkage of the coefficient of multiple correlation. *Journal of Educational Psychology*, 22(1):45.

Lemaire, J.
2013. *Automobile insurance: actuarial models*, volume 4. Springer Science & Business Media.

Leonardi, F. et al.
2010. Some upper bounds for the rate of convergence of penalized likelihood context tree estimators. *Brazilian Journal of Probability and Statistics*, 24(2):321–336.

Levikson, B. and G. Mizrahi
1994. Pricing long term care insurance contracts. *Insurance: Mathematics and Economics*, 14(1):1–18.

Liesen, J. and V. Mehrmann
2015. Vector Spaces. In *Linear Algebra*, Pp. 115–133. Springer.

Lin, S.-K. et al.
2008. Pricing catastrophe insurance products in Markov jump diffusion models. *Journal of Financial Studies*, (16).

Little, R. J. and D. B. Rubin
2019. *Statistical analysis with missing data*, volume 793. John Wiley & Sons.

Longley-Cook, L. H.
1961. Insurance: Its Theory and Practice in the United States. *Journal of the Institute of Actuaries*, 87(3):403–405.

Mächler, M. and P. Bühlmann
2004. Variable length Markov chains: methodology, computing, and software. *Journal of Computational and Graphical Statistics*, 13(2):435–455.

Mann, H. B. and A. Wald
1943. On stochastic limit and order relationships. *The Annals of Mathematical Statistics*, 14(3):217–226.

Markov, A. A.
1906. Extension of the law of large numbers to dependent quantities. *Izv. Fiz.-Matem. Obsch. Kazan Univ.(2nd Ser)*, 15:135–156.

Metzler, R.
2000. Generalized Chapman-Kolmogorov equation: A unifying approach to the description of anomalous transport in external fields. *Physical Review E*, 62(5):6233.

Microsoft Corporation and S. Weston
2019. doParallel: Foreach Parallel Adaptor for the 'parallel' package. R package version 1.0.15.

Milbrodt, H. and A. Stracke
1997. Markov models and Thiele's integral equations for the prospective reserve. *Insurance: Mathematics and Economics*, 19(3):187–235.

Morris, C. N.
1983. Parametric empirical Bayes inference: theory and applications. *Journal of the American Statistical Association*, 78(381):47–55.

Norberg, R.
1991. Reserves in life and pension insurance. *Scandinavian Actuarial Journal*, 1991(1):3–24.

Norberg, R.
2014. Life insurance mathematics. *Wiley StatsRef: Statistics Reference Online*, Pp. 1–19.

Pachet, F.
2002. Playing with virtual musicians: The continuator in practice. *IEEE MultiMedia*, 9(3):77–82.

Papoulis, A. and S. U. Pillai
2002. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education.

Phelan, M. J.
1988. Inference from censored Markov chains with applications to multiwave panel data. *Stochastic Processes and their Applications*, 29(1):85–102.

Pinsker, M. S.
1973. On the complexity of a concentrator. In *7th International Telegraffic Conference*, volume 4, Pp. 1–318. Citeseer.

R Core Team
2018. R: A language and environment for statistical computing.

Rieder, H.
2012. *Robust asymptotic statistics*, volume 1. Springer Science & Business Media.

Rissanen, J.
1983. A universal data compression system. *IEEE Transactions on Information Theory*, 29(5):656–664.

Ron, D., Y. Singer, and N. Tishby
1996. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, 25(2-3):117–149.

Rotar, V. I.
2014. *Actuarial models: the mathematics of insurance*. CRC Press.

Ruckdeschel, P. and H. Rieder
2004. Optimal influence curves for general loss functions. *Statistics & Decisions/International mathematical journal for stochastic methods and models*, 22(3/2004):201–223.

Schmidt, K. D.
2006. *Versicherungsmathematik*. Springer-Verlag.

Schwarz, G. et al.
1978. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

SIGNAL IDUNA
2020. SIGNAL IDUNA. `https://www.signal-iduna.de/`, Webpage accessed: 2020-07-17.

Sohrab, H. H.
2003. *Basic real analysis*, volume 231. Springer.

Statistisches Bundesamt
2018. Pflegestatistik. `https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Pflege/Publikationen/Downloads-Pflege/pflege-deutschlandergebnisse-5224001179004.pdf`, Webpage accessed: 2020-07-15.

Tappe, S.
2013. *Einführung in die Wahrscheinlichkeitstheorie*. Springer.

Tibshirani, R.
1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Van der Vaart, A. W.
2000. *Asymptotic statistics*, volume 3. Cambridge University Press.

Verbelen, R., K. Antonio, and G. Claeskens
2018. Unravelling the predictive power of telematics data in car insurance pricing. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67(5):1275–1304.

Wang, S. S.
2002. A universal framework for pricing financial and insurance risks. *ASTIN Bulletin: The Journal of the IAA*, 32(2):213–234.

Weinberger, M. J., J. J. Rissanen, and M. Feder
1995. A universal finite memory source. *IEEE Transactions on Information Theory*, 41(3):643–652.

Werner, D.
2006. *Funktionalanalysis*. Springer.

Weston, S.
2013. doMPI: Foreach parallel adaptor for the Rmpi package. *R package version 0.2*, 16.

Wickham, H.
2016. Scales: scale functions for visualization. *R package version 0.4*.

Wilder, J.
2002. John Wilder Tukey. *NOTICES OF THE AMS*, 49(2).

Wüthrich, M. V.
2017. Covariate selection from telematics car driving data. *European Actuarial Journal*, 7(1):89–108.

Xie, Y.
2013. knitr: A general-purpose Tool for dynamic report generation in R. *R package version 1.21*, 1(1).

Xiong, J., V. Jääskinen, J. Corander, et al.
2016. Recursive learning for sparse Markov models. *Bayesian analysis*, 11(1):247–263.

Appendices

A Mathematical tools

Some mathematical tools used within this thesis are commonly known. To increase the readability of the actual thesis we did not explicitly write them down earlier. For an easy access for the reader we now catch up on this. Additionally we give references for the statements and/or their proofs.

Theorem A.1 (Strong law of large numbers, Etemadi, 1981)**.** *Let $(X_n)_{n\in\mathbb{N}}$ be a sequence of pairwise independent, identically distributed random variables. Let*

$$S_n := \sum_{i=1}^{n} X_i.$$

Then, if $\mathbb{E}[|X_1|] < \infty$,

$$\frac{1}{n} S_n \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} \mathbb{E}[X_1].$$

A proof of Theorem A.1 is given in the original paper Etemadi (1981, pages 119-120).

Theorem A.2 (Chapman-Kolmogorov equation, Kolmogorov, 1931)**.** *Let $(X_t)_{t\in\mathbb{N}}$ be a time-discrete Markov process of first order on the finite state space M, let $1 \leq s \leq k \leq t$ and $x, w \in M$ with $\mathbb{P}(X_s = w) > 0$. It holds that*

$$\mathbb{P}(X_t = x \,|\, X_s = w) = \sum_{u\in M} \mathbb{P}(X_t = x \,|\, X_k = u)\, \mathbb{P}(X_k = u \,|\, X_s = w).$$

The Chapman-Kolmogorov equation "dates back to Bachelier's treatises of stock market speculation, Smoluchowski's work on colloidal particles, Chapman's studies of the diffusion of grains in a nonuniform fluid and Kolmogorov's probability theoretical investigation.", cf. Metzler (2000, page 6233). For a derivation we refer to Papoulis and Pillai (2002, page 705).

Proposition A.3. *Let M be a set of finite cardinality $m \in \mathbb{N}$. Then the power set $\mathscr{P}(M)$ of M is of cardinality 2^m.*

Cf. e.g. Halmos (2017, page 20).

Proposition A.4 (Sandwich theorem)**.** *Let* $(a_n)_{n\in\mathbb{N}} \subseteq \mathbb{R}$ *and* $(c_n)_{n\in\mathbb{N}} \subseteq \mathbb{R}$ *be two sequences with* $a_n \xrightarrow{n\to\infty} b$ *and* $c_n \xrightarrow{n\to\infty} b$*. If* $(b_n)_{n\in\mathbb{N}} \subseteq \mathbb{R}$ *is a third sequence with*

$$a_n \leq b_n \leq c_n$$

for all n *larger than a threshold* $n_0 \in \mathbb{N}$*, it must be that* $b_n \xrightarrow{n\to\infty} b$*.*

Proposition A.4 is often also called "squeeze theorem" or "pinching theorem" and e.g. can be found under Theorem 3.3.6 in Sohrab (2003, page 104).

Lemma A.5 (Dragomir et al., 2000)**.** *Let* P *and* Q *be two discrete probability distributions on* Ω *with* $P(x) > 0$ *and* $Q(x) > 0$ *for all* $x \in \Omega$*. Then*

$$\mathrm{D}\left[P||Q\right] \leq \left(\sum_{x\in\Omega} \frac{P^2(x)}{Q(x)}\right) - 1.$$

Lemma A.5 can be found under Theorem 1 in Dragomir et al. (2000, page 94). A proof is also given in aforesaid reference.

Theorem A.6 (Hoeffding's inequality, Hoeffding, 1963)**.** *Let* $(X_n)_{n\in\mathbb{N}}$ *be independent random variables bounded almost surely by the interval* $[0, 1]$*. Then*

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right]\right| \geq t\right) \leq 2e^{-2nt^2}$$

holds for $t \geq 0$*.*

For a proof cf. the proof of Theorem 1 in Hoeffding (1963, pages 20-22).

Corollary A.7 (Hoeffding's inequality)**.** *Let* $(X_n)_{n\in\mathbb{N}}$ *be independent random variables bounded almost surely by the interval* $[0, 1]$*. Then*

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] \geq t\right) \leq e^{-2\frac{t^2}{n}}$$

and

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] \leq -t\right) \leq e^{-2\frac{t^2}{n}}$$

holds for $t \geq 0$*.*

This is Theorem 2.16 in Bercu et al. (2015, pages 21-23). A proof can also be found there. In particular pay attention to the equations (2.35) and (2.36).

Theorem A.8 (Law of the iterated logarithm, Khintchine, 1924)**.** *Let* $(X_n)_{n\in\mathbb{N}}$ *be independent and identically distributed random variables with* $\mathbb{E}[X_1] = 0$ *and* $\mathrm{Var}[X_1] = 1$*. Let*

$$S_n := \sum_{i=1}^{n} X_i.$$

Then

$$\limsup_{n\to\infty} \frac{S_n}{\sqrt{2n \log\log n}} = 1$$

holds $\mathbb{P}$*-a.s.*

Theorem A.8 is proven e.g. in the original paper Khintchine (1924, pages 10-19) or (written in a more modern fashion) in the proof of Theorem 9.5 in Billingsley (1995, pages 154-155).

Definition A.9 (Bachmann-Landau notation)**.** Let $(X_n)_{n\in\mathbb{N}}$ and $(Y_n)_{n\in\mathbb{N}}$ be two sequences of random variables. We write $X_n = \mathcal{O}(Y_n)$ if

$$\limsup_{n\to\infty} \left|\frac{X_n}{Y_n}\right| < \infty$$

holds $\mathbb{P}$-a.s.

The Bachmann-Landau notation is also often referred to as "big O notation". It was invented by Paul Bachmann, one of its first appearances being Bachmann (1904).

Theorem A.10 (Norm equivalence, Werner, 2006)**.** *If* $\|\cdot\|$ *and* $\|\cdot\|_*$ *are two norms on a vector space* V*, the following two statements are equivalent:*

(a) $\|\cdot\|$ *and* $\|\cdot\|_*$ *are equivalent.*

(b) A sequence converges with regards to $\|\cdot\|$ *if and only if it converges with regards to* $\|\cdot\|_*$ *and both limits are the same.*

Furthermore, if V *is of finite dimension, all norms on* V *are equivalent.*

For a proof of Theorem A.10 e.g. cf. Werner (2006, pages 24-26).

Theorem A.11 (Continuous mapping theorem, Mann and Wald, 1943)**.** *Let $(X_n)_{n\in\mathbb{N}}$ and X be random variables defined on the same probability space. Let g be continuous at every point of a set C such that $\mathbb{P}(X \in C) = 1$, then*

$$X_n \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} X \Rightarrow g(X_n) \xrightarrow[\mathbb{P}-\text{a.s.}]{n\to\infty} g(X).$$

A proof is given in the original paper Mann and Wald (1943), but a more compressed and modern source is Van der Vaart (2000, page 8), Theorem 2.3.

Proposition A.12 (Bayes' theorem)**.** *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $(B_i)_{i\in I}$ a partition of Ω. Then*

$$\mathbb{P}(A) = \sum_{i\in I} \mathbb{P}(A \mid B_i)\mathbb{P}(B_i)$$

holds for all $A \in \mathcal{F}$.

The famous Bayes' theorem A.12 is part of every undergraduate course in stochastic. A proof can be found e.g. in Tappe (2013, page 26) or in Chapter 8.7 in Kendall et al. (1948).

Proposition A.13 (De Morgan's laws)**.** *Let $(B_i)_{i\in I}$ be a family of subsets of Ω. Then*

$$\Big(\bigcup B_i\Big)^C = \bigcap B_i^C$$

and

$$\Big(\bigcap B_i\Big)^C = \bigcup B_i^C$$

hold.

For a proof cf. the proof of Lemma 2.2 in Tappe (2013, page 8).

B Context tree estimates (data.signal)

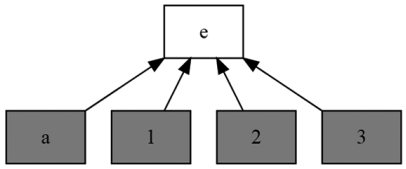

Figure B.1: Context tree number 1 of the 43 occurring trees (data.signal)

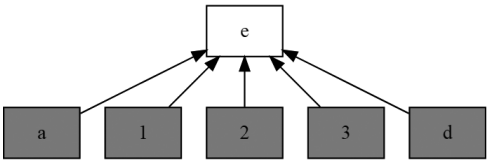

Figure B.2: Context tree number 2 of the 43 occurring trees (data.signal)

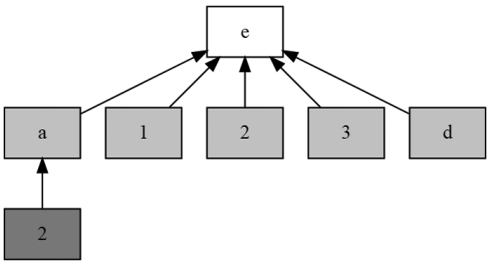

Figure B.3: Context tree number 3 of the 43 occurring trees (data.signal)

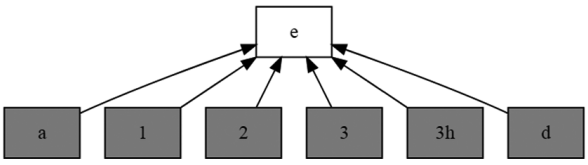

Figure B.4: Context tree number 4 of the 43 occurring trees (data.signal)

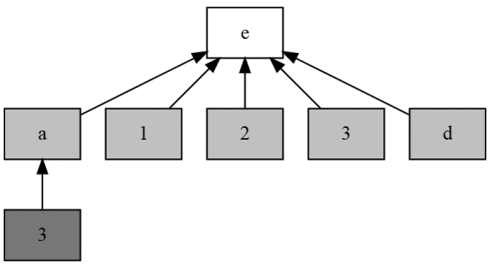

Figure B.5: Context tree number 5 of the 43 occurring trees (data.signal)

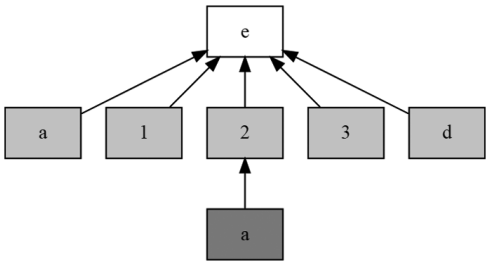

Figure B.6: Context tree number 6 of the 43 occurring trees (data.signal)

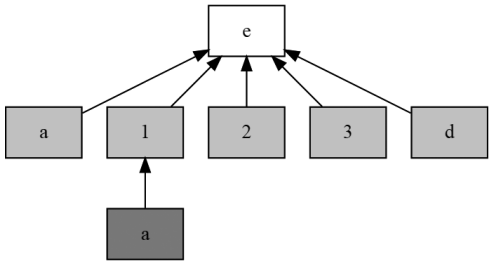

Figure B.7: Context tree number 7 of the 43 occurring trees (data.signal)

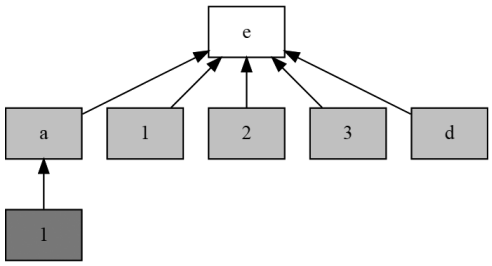

Figure B.8: Context tree number 8 of the 43 occurring trees (data.signal)

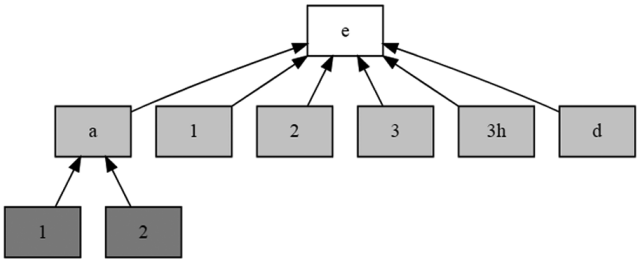

Figure B.9: Context tree number 9 of the 43 occurring trees (data.signal)

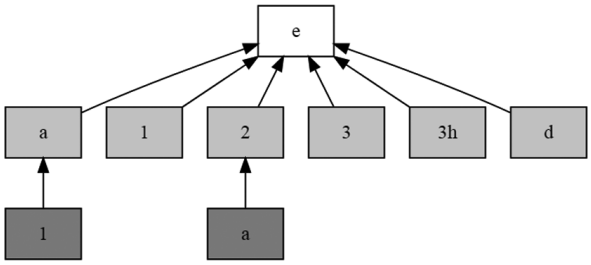

Figure B.10: Context tree number 10 of the 43 occurring trees (data.signal)

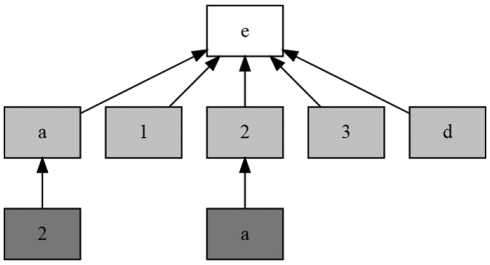

Figure B.11: Context tree number 11 of the 43 occurring trees (data.signal)

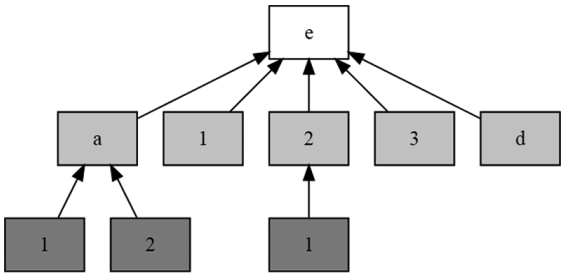

Figure B.12: Context tree number 12 of the 43 occurring trees (data.signal)

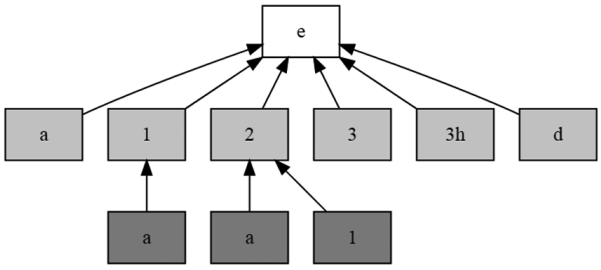

Figure B.13: Context tree number 13 of the 43 occurring trees (data.signal)

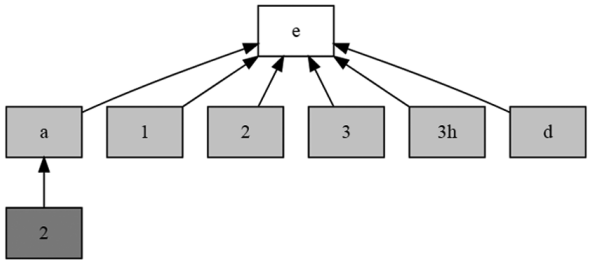

Figure B.14: Context tree number 14 of the 43 occurring trees (data.signal)

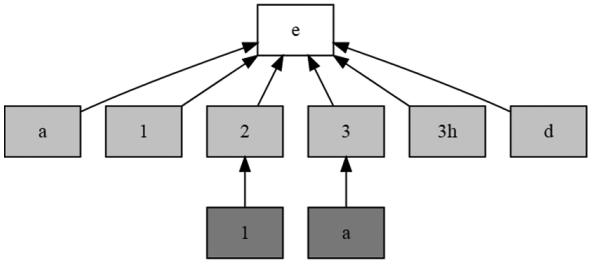

Figure B.15: Context tree number 15 of the 43 occurring trees (data.signal)

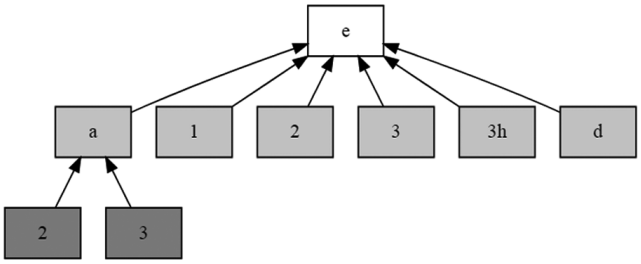

Figure B.16: Context tree number 16 of the 43 occurring trees (data.signal)

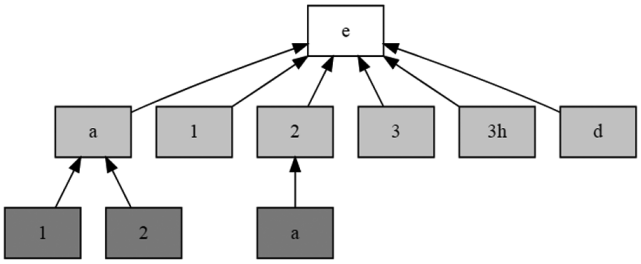

Figure B.17: Context tree number 17 of the 43 occurring trees (data.signal)

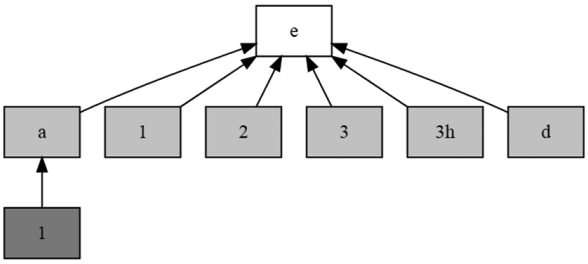

Figure B.18: Context tree number 18 of the 43 occurring trees (data.signal)

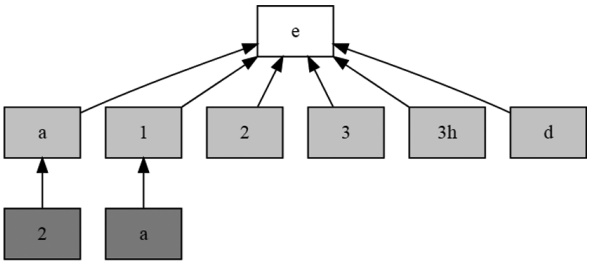

Figure B.19: Context tree number 19 of the 43 occurring trees (data.signal)

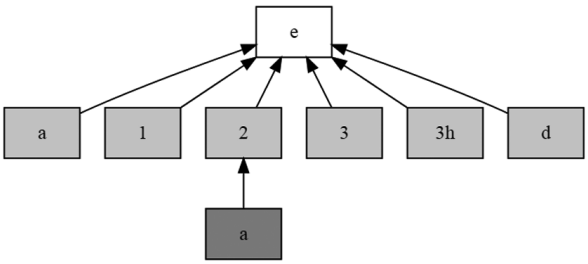

Figure B.20: Context tree number 20 of the 43 occurring trees (data.signal)

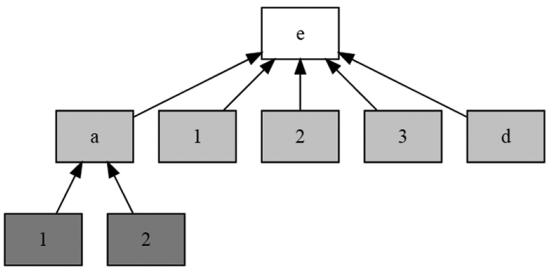

Figure B.21: Context tree number 21 of the 43 occurring trees (data.signal)

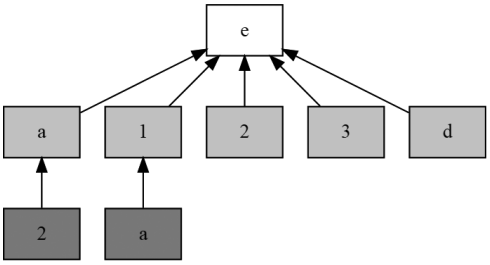

Figure B.22: Context tree number 22 of the 43 occurring trees (data.signal)

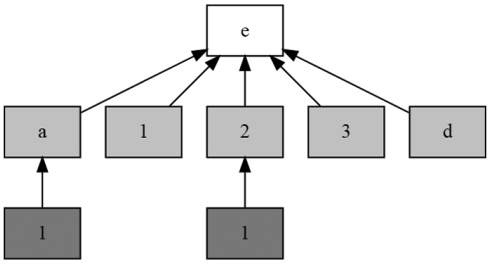

Figure B.23: Context tree number 23 of the 43 occurring trees (data.signal)

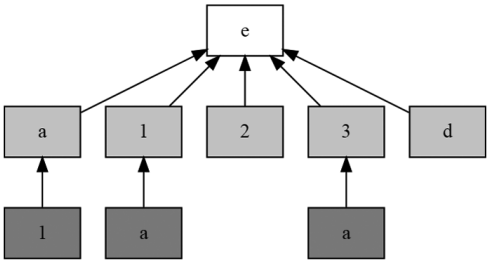

Figure B.24: Context tree number 24 of the 43 occurring trees (data.signal)

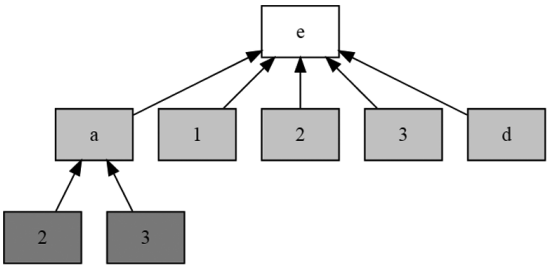

Figure B.25: Context tree number 25 of the 43 occurring trees (data.signal)

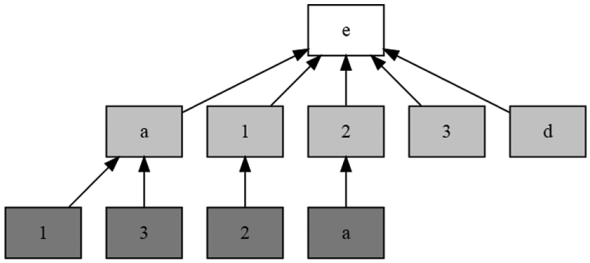

Figure B.26: Context tree number 26 of the 43 occurring trees (data.signal)

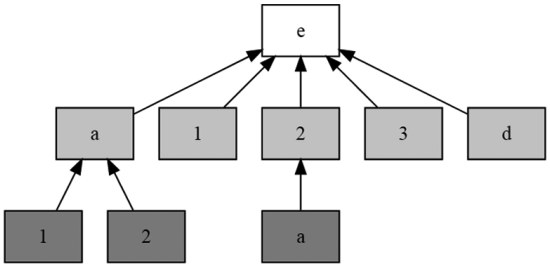

Figure B.27: Context tree number 27 of the 43 occurring trees (data.signal)

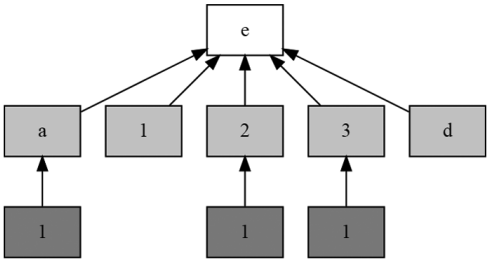

Figure B.28: Context tree number 28 of the 43 occurring trees (data.signal)

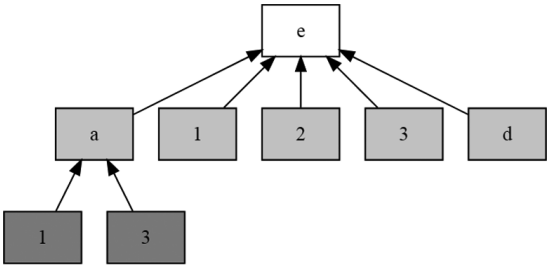

Figure B.29: Context tree number 29 of the 43 occurring trees (data.signal)

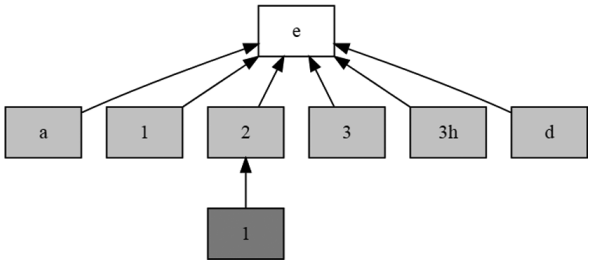

Figure B.30: Context tree number 30 of the 43 occurring trees (data.signal)

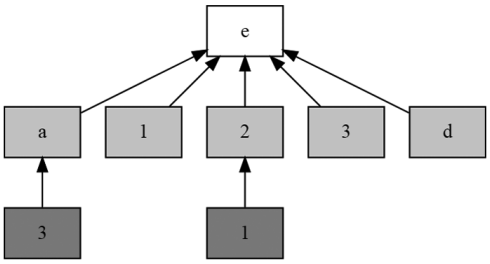

Figure B.31: Context tree number 31 of the 43 occurring trees (data.signal)

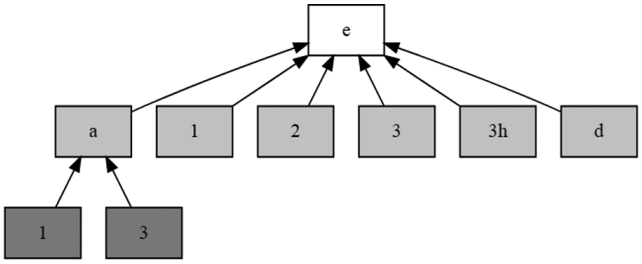

Figure B.32: Context tree number 32 of the 43 occurring trees (data.signal)

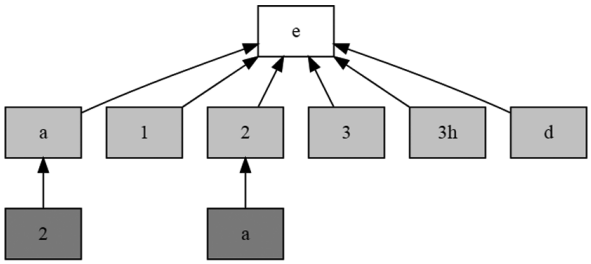

Figure B.33: Context tree number 33 of the 43 occurring trees (data.signal)

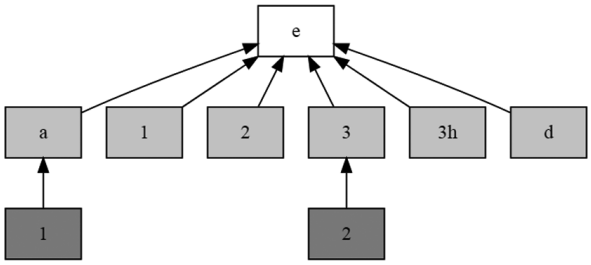

Figure B.34: Context tree number 34 of the 43 occurring trees (data.signal)

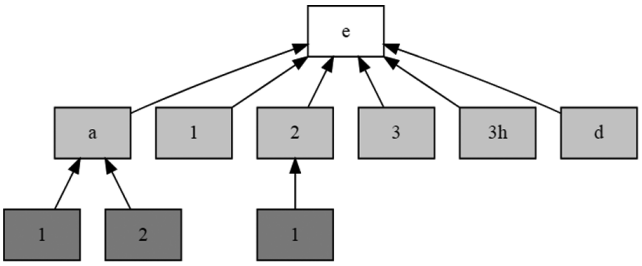

Figure B.35: Context tree number 35 of the 43 occurring trees (data.signal)

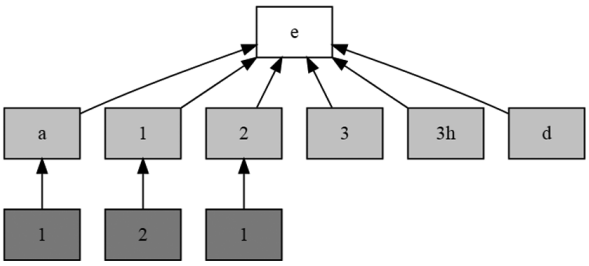

Figure B.36: Context tree number 36 of the 43 occurring trees (data.signal)

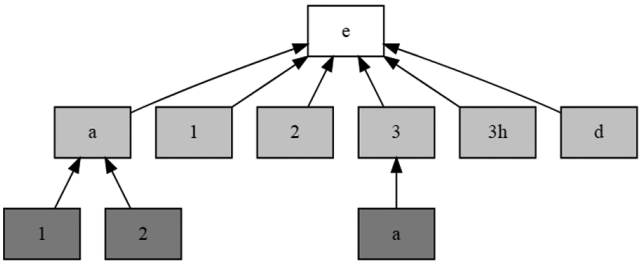

Figure B.37: Context tree number 37 of the 43 occurring trees (data.signal)

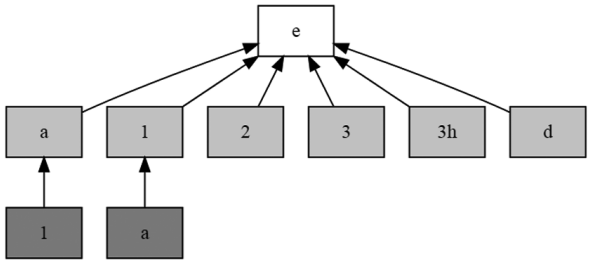

Figure B.38: Context tree number 38 of the 43 occurring trees (data.signal)

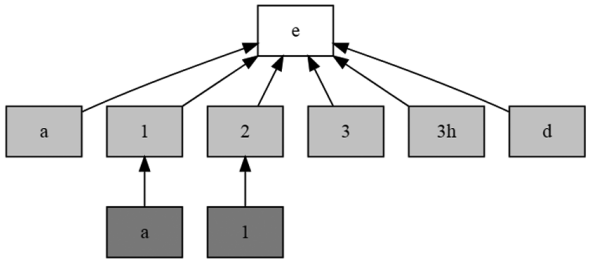

Figure B.39: Context tree number 39 of the 43 occurring trees (data.signal)

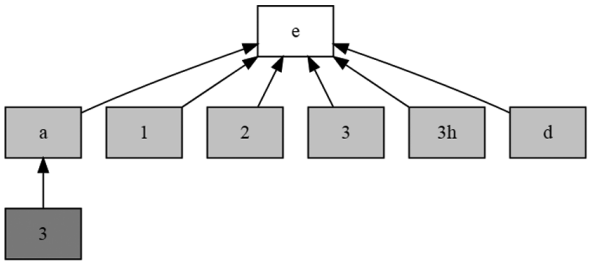

Figure B.40: Context tree number 40 of the 43 occurring trees (data.signal)

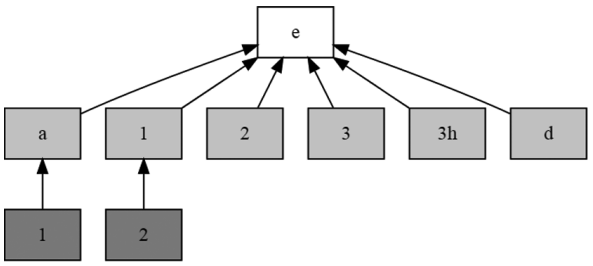

Figure B.41: Context tree number 41 of the 43 occurring trees (data.signal)

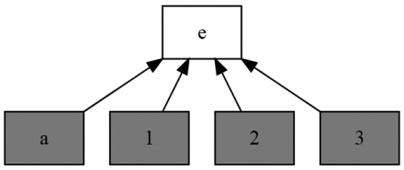

Figure B.42: Context tree number 42 of the 43 occurring trees (data.signal)

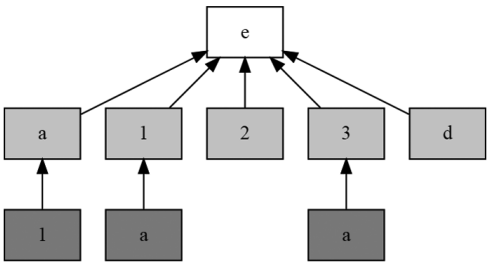

Figure B.43: Context tree number 43 of the 43 occurring trees (data.signal)

C Context tree estimates (data.female)

Figure C.1: Context tree number 1 of the 14 occurring trees (data.female)

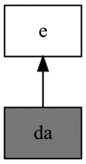

Figure C.2: Context tree number 2 of the 14 occurring trees (data.female)

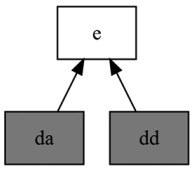

Figure C.3: Context tree number 3 of the 14 occurring trees (data.female)

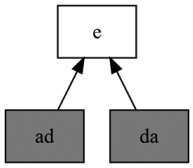

Figure C.4: Context tree number 4 of the 14 occurring trees (data.female)

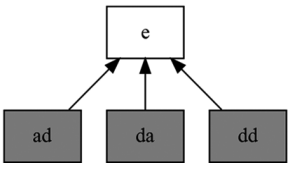

Figure C.5: Context tree number 5 of the 14 occurring trees (data.female)

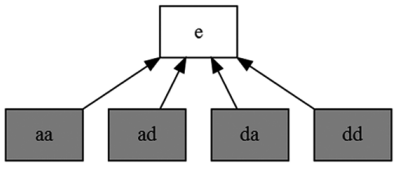

Figure C.6: Context tree number 6 of the 14 occurring trees (data.female)

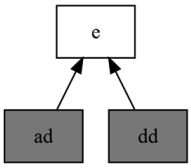

Figure C.7: Context tree number 7 of the 14 occurring trees (data.female)

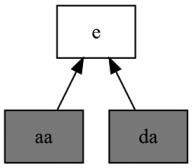

Figure C.8: Context tree number 8 of the 14 occurring trees (data.female)

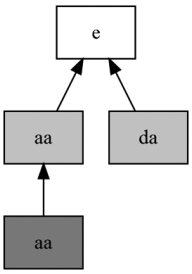

Figure C.9: Context tree number 9 of the 14 occurring trees (data.female)

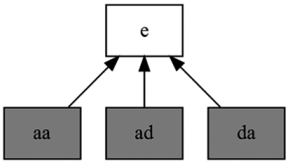

Figure C.10: Context tree number 10 of the 14 occurring trees (data.female)

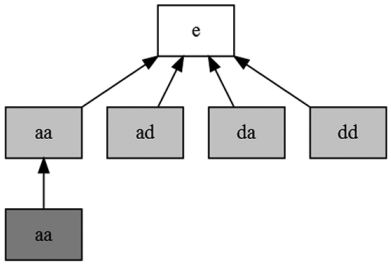

Figure C.11: Context tree number 11 of the 14 occurring trees (data.female)

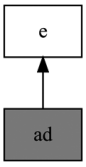

Figure C.12: Context tree number 12 of the 14 occurring trees (data.female)

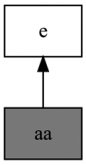

Figure C.13: Context tree number 13 of the 14 occurring trees (data.female)

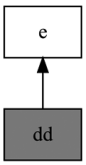

Figure C.14: Context tree number 14 of the 14 occurring trees (data.female)

Curriculum Vitae

3rd July 1993	born in Bremen, Germany
August 2004 - July 2012	High school student Max-Planck-Gymnasium, Delmenhorst, Germany graduated with Abitur (secondary school diploma)
October 2012 - July 2015	Undergraduate studies in mathematics University of Oldenburg, Oldenburg, Germany graduated with a BSc in mathematics
October 2014 - March 2015	Student assistant at the Institute of Mathematics University of Oldenburg, Oldenburg, Germany
October 2015 - September 2017	Postgraduate studies in mathematics University of Oldenburg, Oldenburg, Germany graduated with a MSc in mathematics
October 2015 - September 2016	Scholarship "Deutschlandstipendium" awarded by the University of Oldenburg to honour outstanding academical performance
August 2016 - September 2016	Intern KROSE GmbH & Co. KG, Bremen, Germany
October 2016 - September 2017	Scholarship "Deutschlandstipendium" awarded by the University of Oldenburg to honour outstanding academical performance
October 2017 - present	Research assistant at the Institute of Mathematics University of Oldenburg, Oldenburg, Germany
October 2017 - present	Doctoral studies in mathematics University of Oldenburg, Oldenburg, Germany
February 2018	Award "Preis des Instituts für Mathematik" awarded by the University of Oldenburg for an outstanding master's thesis

Affidavit

I hereby declare that the submitted thesis entitled "Finding Markovian Models for Insurance Processes by Expanding State Spaces" is my own work. I have only used the sources indicated and have not made unauthorised use of services of a third party. Where the work of others has been quoted or reproduced, the source is always given.
I further declare that the submitted thesis or parts thereof have not been presented as part of an examination degree to any other university.