Measuring credit rating downgrade momentum

a study on parameter estimation methodologies for non-Markovian models

by

R.W.W. Sluijmer

To obtain a degree of Master of Science
at Delft University of Technology,
To be defended publicly on Friday February 11, 2022 at 15:00.

Student number:	4292685	
Project duration:	from February 1 2021 to February 1 2022	
Thesis committee:	Dr. N. Parolya	TU Delft, daily supervisor
	Dr. A. Papapantoleon	TU Delft

Abstract

In the pursue of accurately computing default probabilities of financial instruments, external credit ratings, which are issued by credit rating agencies, have been commonly modelled by continuous-time Markov chains [21] [22] [30]. The stochastic behaviour of these ratings, $\boldsymbol{X}$, is driven by a generator matrix, $\boldsymbol{Q}$, which can be approximated using either discrete and anonymous or continuous rating data. Parameter estimation methodologies for both types of data have been optimized in past literature allowing for confidence intervals in the resulting default probabilities [21] [22] [30] [37]. As there is strong evidence of non-Markovian behaviour among credit ratings with downgrade momentum being dominant, a new non-Markovian model is described in recent research modelling this form of momentum [4] [10] [15] [21] [37] [40] [41] [43]. As opposed to past non-Markovian models, recent research successfully captures momentum accumulating and decaying over time using a parsimonious model [9] [14] [15] [16] [22] [25] [36]. Recent research hypothesized that the parameters, $\boldsymbol{\theta}$, could only be estimated by a Metropolis-Hastings algorithm avoiding complex first and/or second derivatives of the loglikelihood, $l(\boldsymbol{\theta}|\boldsymbol{X})$, which required 8.5 hours of computational time [21]. This research has introduced a new successful parameter estimation methodology according to a maximum likelihood estimator based (projected) Newton-Raphson method and explored several alternative models and estimators: a modified Markovian model, heuristic estimator (based on Chapman-Kolmogorov equations) and discretized simulation [5] [24] [58]. The performances of the parameter estimation methodologies, alternative models and estimators were also back-tested on simulated data using parameter estimations found in recent research [13] [19] [21]. Among the alternative models and estimators, only the modified Markovian model has produced reasonable results by approximating non-Markovian default probabilities. Applying a Markovian model based on realizations with the same initial state and therefore similar momentum reduces the intertwining of ratings with significant or none non-Markovian stochastic behaviour allowing for the Markovian model to be an appropriate model in the setting of this research. The (projected) Newton-Raphson method has decreased computational time required for parameter estimation of the non-Markovian model from 6 to 1 hour. Moreover, the maximum likelihood estimator $\hat{\boldsymbol{\theta}}$ according to the projected Newton-Raphson method is a strong initial guess for further parameter estimations based on any sufficiently large subsamples of the realizations. This allows for computing confidence intervals of the non-Markovian default probabilities via parametric bootstrapping as the parameter estimations of subsamples require significantly less computational time. Further research can be done on the reduction of the computational time required allowing for more precise confidence intervals of default probabilities per initial state via parametric bootstrapping based on more subsamples. Also, the method for discretized simulation could be reviewed investigating the possibility of disregarding certain simulated paths as their overall contribution to default probabilities is negligible. Finally, model expansions incorporating correlation, upgrade momentum, more granularity and business cycles can be considered in future non-Markovian models.

Contents

1 Introduction

1.1 Probability of default

At the heart of pricing or measuring risk of any financial instrument, e.g., companies' equity, consumer loans or complex derivatives, lies the computation of its probability of default (PD) over time [56]. Any participator in the financial market is exposed to some form of such a (credit risk) assessment, however, in the highly regulated banking industry correct credit risk modelling is considered of vital importance [56]. During the global financial crises of '07-'08, the significance of accurate credit risk modelling was highlighted and is currently emphasized by ongoing implementations of the revised Basel framework (Basel IV) and reformed International Financial Reporting Standard (IFRS-9) [3] [21]. These regulations demand computations of PDs for financial instruments over longer time horizons, including their complete lifetimes, by banks all over the world [21]. Besides an apparent demand for more advanced mathematics, increasing data availability continues to drive research on further developed methodologies in the financial market.

In general, there are two ways of determining PDs: implied by current market valuations (risk-neutral) or based on historical data (real-world) [3] [6] [21]. There are various prominent credit risk methodologies using historical data each with their own pros and cons from a mathematical and economic point of view [3].

Definitions real-world and risk-neutral probability: real-world (physical) PDs are based on historical data and open to interpretation by various models [3] [6][21], where (unique) risk-neutral PDs are implied by current market valuations under the (strong) assumptions that markets are arbitrage-free and complete with the relevant financial instruments being priced according to an universal model [21] [47].

This research expands on past literature that focused on computing real-world PDs based on widely used external credit ratings [21].

1.2 External credit ratings

Credit ratings of financial instruments (hereinafter referred to as "ratings") are labels used as (finite) categorical measures that allow financial instruments to be ordered based on their PDs (or solvency in general) [21]. Ratings are externally issued by credit rating agencies (CRAs) or internally by banks themselves according to the Basel regulatory framework [21]. External ratings belong to the most widely used credit risk methodologies for banks all over the world [3]. Prior to the application by banks, CRAs have based these ratings on proprietary models with a wide range of drivers, e.g., environmental conditions, competitive positions, management quality and financial strengths [3]. Banks benefit from this as external ratings are ready to use, while evolving from complex analyses by CRAs. These ratings are, however, also considered as relative and subjective measures, which are not continuously monitored and therefore allowing for deviations from the real underlying rating throughout time [3].

1.3 Credit rating transitions

Because of changing underlying drivers over time according to the proprietary models of CRAs, external ratings are dynamic [21]. The issuance of a different rating for a financial instrument (by the same CRA) is called a credit rating transition (hereinafter referred to as a "transition") [21]. Consequently, a finite number of possible ratings create a discrete state space, $\boldsymbol{S}$, including a default state, h. Ratings of financial instruments make transitions between these states over time, which naturally leads to the construction of a continuous-time Markov chain model (CTMC). According to a CTMC model the Markovian property is assumed, which implies that transition probabilities are solely dependent on the current rating of a financial instrument [26].

Definition Markovian property: a stochastic process (rating) is said to be Markovian (or satisfy the Markovian property) if event probabilities (transitions) only depend on its current state creating a Markov chain [26]. Non-Markovian behaviour implies that past states (ratings) influence current transition probabilities [21].

1.4 Downgrade momentum

Past literature has shown significant statistical presence of non-Markovian behaviour among ratings in datasets of various CRAs and also identified economic drivers for this phenomena [4] [10] [15] [21] [37] [40] [41] [43]. Among different non-Markovian behaviours, credit rating downgrade momentum (hereinafter referred to as "momentum") is considered to be dominant [21] [37]. Momentum implies that financial instruments, which have been downgraded in the past, are increasingly likely to be downgraded further in the future. It has been shown that this effect accumulates and decays over time [15] [21]. This effect is claimed to be less apparent for upgrades and therefore not further researched [4] [15] [21]. Considering the highly regulated and increasingly prudent nature of the global banking industry, it is essential to accurately incorporate momentum in credit risk modelling as past literature has shown its implications become more pronounced over time and may significantly affect long-term PDs [15] [21] [56].

1.5 Literature study

Based on discrete and anonymous (missing) or continuous (full) external credit rating transition data, various methodologies have been used to construct CTMC models via maximum likelihood estimators (MLE). The "classic" problem using missing data has been thoroughly researched and optimized via the estimation of a generator matrix, $\boldsymbol{Q}$, using an expectation-maximization (EM) algorithm [7] [8] [21] [22] [30] [45] [48]. Estimating $\boldsymbol{Q}$ via MLE using full data is relatively straightforward [21] [30] [33] [37].

Past literature has also incorporated non-Markovian behaviour of ratings using relatively less straightforward constructions like extended state spaces, mixture models, hidden Markov models or semi-Markov models, focusing on momentum [9] [14] [16] [25] [36], but failed to incorporate the accumulating behaviour decaying over time [15] [21]. Recent research is based on exponential Hawkes marked point processes (EHMPP) and successfully incorporated accumulating momentum decaying over time, while allowing for granularity in the momentum contribution per rating using a parsimonious model [21]. Consequently, due to an increase in model complexity (following recent research) compared to classical CTMC models, there is an opportunity to back-test and possibly optimize current parameter estimation methodologies. In addition, alternative methodologies for computing PDs incorporating momentum to overcome certain computational requirements can be explored.

In short, this report expands on recent research, which describes a non-Markovian model incorporating momentum applied to credit ratings and is focused on the following three topics [21]:

1. Back-testing current parameter estimation methodologies on simulated data;
2. Introducing a new parameter estimation methodology focused on better performance with regards to (i) robustness, (ii) accuracy and (iii) efficiency;
3. Exploring alternative methodologies for accurately computing PDs incorporating momentum.

2 Theory

2.1 Markovian model

First, let $\boldsymbol{X}(t) = \{X(t,\omega) : [0,T] \times \boldsymbol{\Omega} \to \boldsymbol{S}\}$ be a right-continuous stochastic process defined on some probability space $(\boldsymbol{\Omega}, \boldsymbol{\mathcal{F}}_t, \mathbb{P})$ taking on values in some measurable finite state space $(\boldsymbol{S}, \boldsymbol{\Sigma})$ over time horizon $[0,T]$ such that $t \in [0,T]$ with $T \in \mathbb{R}^+$, $\mathcal{F}_t$ the natural filtration, $\boldsymbol{S} = \{1,...,h\}$, $\Sigma = \sigma(\boldsymbol{S})$ and $h \in \mathbb{N}^+$ under the probability measure $\mathbb{P} : \boldsymbol{\mathcal{F}}_t \to [0,1]$. $\boldsymbol{X}(t)$ describes a rating with a defaulting possibility modelled by a finite state CTMC with an absorption state, h [21] [22] [30].

Second, assume $\boldsymbol{X}(t)$ satisfies the Markovian property implying memorylessness such that the probability measure, $\mathbb{P}$, only depends on the current state of $\boldsymbol{X}(t)$ for any $i \in \boldsymbol{S}$ as follows [30] (5.2.1.1)

$$\mathbb{P}(X(t+\Delta t) = i|\mathcal{F}_t) = \mathbb{P}(X(t+\Delta t) = i|X_t)$$

Third, assume $\boldsymbol{X}(t)$ is time-homogeneous implying $\mathbb{P}$ is constant over time horizon $[0,T]$ for any $i,j \in \boldsymbol{S}$, $\Delta t > 0$ and $t_1, t_2 \in [0,T]$ such that $t_1 + \Delta t, t_2 + \Delta t \in [0,T]$ as follows [30] (5.2.1.2)

$$\begin{aligned}&\mathbb{P}(X(t_1+\Delta t) = j|X(t_1) = i)\\ &\qquad = \mathbb{P}(X(t_2+\Delta t) = j|X(t_2) = i)\end{aligned}$$

Both assumptions are commonly used in past literature to describe CTMCs applied to ratings [30]. Consequently, these assumptions imply that for any state $i \in \boldsymbol{S}$ the holding times, T_i, are independent and identically distributed (IID) exponential random variables with parameter $q_i > 0$ as follows [30] [34] [39] [44] [52] (5.2.1.3)

$$f_{T_i}(x) = q_i e^{-q_i x}$$

such that the holding probability, $\mathbb{P}(T_i > \Delta t)$, for any Δt and $t \in [0,T]$ with $t + \Delta t \in [0,T]$ is defined by [1] (5.2.1.4)

$$\begin{aligned}\mathbb{P}(T_i > \Delta t) &= \mathbb{P}(\{X(\bar{t}) = i : \bar{t} \in [t, t+\Delta t]\}|X(t) = i)\\ &= e^{-q_i \Delta t}\end{aligned}$$

Correspondingly, the number of transitions, $N(\Delta t)$, out of state $i \in \boldsymbol{S}$ over a time interval with length Δt is Poisson distributed as follows [1] [27] [30] [54] (5.2.1.4)

$$\mathbb{P}(N(\Delta t) = k) = \frac{(q_i \Delta t)^k}{k!} e^{q_i \Delta t}$$

such that the probability of a single transition out of state $i \in \boldsymbol{S}$, $\mathbb{P}(X(t+\Delta t) = i|X(t) \neq i)$, for any $t \in [0,T]$ and $\Delta t > 0$ with $t + \Delta t \in [0,T]$ is defined by [1] [30] [35] [44] (5.2.1.4)

$$\mathbb{P}(X(t+\Delta t) \neq i|X(t) = i) = \lim_{\Delta t \to 0} q_i \Delta t + \mathcal{O}(\Delta t)$$

Next, define the conditional transition probability parameters, q_{ij}, by the following limit for any $i,j \in \boldsymbol{S}$ with $j \neq i$ [30] [35] (5.2.1.5)

$$q_{ij} = \lim_{\Delta t \to 0} \frac{\mathbb{P}(X(t+\Delta t) = j|X(t) = i)}{\Delta t} \in \mathbb{R}^+$$

such that according to Bayesian theorem the conditional transition probability, $\mathbb{P}(X(t+\Delta t) = j|X(t+\Delta t) \neq i, X(t) = i)$, is given by [30]

$$\begin{aligned}\mathbb{P}(X(t+\Delta t) &= j|X(t+\Delta t) \neq i, X(t) = i)\\ &= \frac{\mathbb{P}(X(t+\Delta t) = j|X(0) = i)}{\mathbb{P}(X(t+\Delta t) \neq i|X(0) = i)}\\ &= \lim_{\Delta t \to 0} \frac{q_{ij}\Delta t + \mathcal{O}(\Delta t)}{q_i \Delta t + \mathcal{O}(\Delta t)}\\ &= \frac{q_{ij}}{q_i}\end{aligned}$$

with $\mathcal{O}$ the big-O-notation [35]. Furthermore, as the sum of all conditional transition probabilities, $\mathbb{P}(X(t+\Delta t) = j|X(t+\Delta t) \neq i, X(t) = i)$, equals 1 for any $i \in \boldsymbol{S}$ and $j = 1,...,h$ with $j \neq i$, the conservation of probability implies that q_i and q_{ij} are related as follows [30]

$$\sum_{j=1,j\neq i}^{h} \frac{q_{ij}}{q_i} = 1 \implies q_i = \sum_{j=1,j\neq i}^{h} q_{ij}$$

such that the Markovian model is in summary defined as follows for any $i, j \in \boldsymbol{S}$

- holding times, T_i, in state i are IID exponential random variables with parameter $q_i > 0$ [30]
- given a transition out of state i, the conditional transition probability, $\mathbb{P}(X(t+\Delta t) = j | X(t+\Delta t) \neq i, X(t) = i)$, to state j is given by $\frac{q_{ij}}{q_i}$ [30]

Subsequently, a stable conservative generator matrix, $\boldsymbol{Q} \in \mathbb{R}^{h\times h}$, can be constructed as follows [30]

$$\boldsymbol{Q} = \begin{pmatrix} -q_1 & q_{12} & q_{13} & \cdots & q_{1h} \\ q_{21} & -q_2 & q_{23} & \cdots & q_{2h} \\ q_{31} & q_{32} & -q_3 & \cdots & q_{3h} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ q_{h1} & q_{h2} & q_{h3} & \cdots & -q_h \end{pmatrix}$$

such that the following conditions for any $i, j \in \boldsymbol{S}$ are satisfied [30]

- $\sum_{j=1}^{h} q_{ij} = 0$
- $0 \leq -q_{ii} = q_i$
- $0 \leq q_{ij}$ with $i \neq j$

Finally, given $\boldsymbol{Q}$ a transition probability matrix (TPM), $\boldsymbol{P} \in \mathbb{R}^{h\times h}$, is defined by [1] [30] (5.2.1.6)

$$\boldsymbol{P}(t, t+\Delta t) = \lim_{\Delta t\to\infty} \boldsymbol{I} + \boldsymbol{Q}\Delta t + \mathcal{O}(\Delta t)$$

such that $\boldsymbol{Q}$ and $\boldsymbol{P}$ are related for any $t_1, t_2 \in [0, T]$ with $t_1 = t_2 - n \cdot \Delta t$ are related as follows

$$\begin{aligned} \boldsymbol{P}(t_1, t_2) &= \lim_{n\to\infty} (\boldsymbol{I} + \boldsymbol{Q}\Delta t)^n \\ &= \lim_{n\to\infty} (\boldsymbol{I} + \boldsymbol{Q}\frac{t_2 - t_1}{n})^n \\ &= e^{\boldsymbol{Q}(t_2 - t_1)} \end{aligned}$$

with $e^{\cdot} = \lim_{n\to\infty}(\boldsymbol{I} + \frac{\cdot}{n})^n$ an alternative definition of the matrix exponential [21] [22] [30] [55].

2.2 Non-Markovian model

Following strong evidence of momentum in rating data of various CRAs according to past literature [4] [10] [15] [21] [37] [40] [41] [43] recent research has described a model using EHMPPs successfully capturing accumulating momentum decaying over time [21].

First, the likelihood of a single MPP, $L(\lambda, f|\boldsymbol{X})$, for $\boldsymbol{X} = \boldsymbol{X}(T)$ over time horizon $[0, T]$ is defined by [17] [18] [21]

$$L(\lambda, f|\boldsymbol{X}) = \prod_{n=1}^{N(T)} \lambda(t_n) f(\kappa_n|t_n) e^{-\int_0^T \lambda(x)dx}$$

with $N(T)$ the total number of transitions, $\lambda(t)$ the intensity, κ_n a mark and $f(\kappa_n|t_n)$ the marks distribution. Consequently, by setting $\lambda(t)$ as follows

$$\lambda(t_n) = \sum_{i=1}^{h} \mathbb{1}_{\{X(t_n)=i\}} q_i$$

with $f(\kappa_n|t_n)$ given by

$$f(\kappa_n|t_n) = \sum_{i,j=1,j\neq i}^{h} \frac{\mathbb{1}_{\{X(t_n)=i, X(t_n^+)=j\}} q_{ij}}{q_i}$$

and a $i \to j$ transition at t_n marked by κ_n the Markovian model is again defined [21] [30] (5.2.2.1).

Second, the intensity $\lambda(t)$ according to the non-Markovian model is defined by an exponential Hawkes process as follows [21]

$$\lambda(t) = \sum_{i=1}^{h-1} \mathbb{1}_{\{X(t)=i\}} q_i + \sum_{\tau\in\boldsymbol{\tau}(t)} \alpha e^{-\beta(t-\tau)}$$

with $\boldsymbol{\tau}(t) = \{\bar{t} : \bar{t} \in [0, t) \wedge X(\bar{t}^+) > X(\bar{t})\}$ the set of past downgrade times up to t and α the magnitude of the intensity impulse, which exponentially decays with rate β over time. Making use of recursive patterns $\lambda(t)$ is given by [21] [46] (5.2.2.2)

$$\lambda(t_n) = \sum_{i=1}^{h-1} \mathbb{1}_{\{X(t_n)=i\}} q_i + \alpha\mathcal{R}_n$$

with the non-Markovian intensity contribution, $\alpha\mathcal{R}_n = \alpha\mathcal{R}(t_n)$, defined by

$$\alpha\mathcal{R}_n = \alpha(\mathcal{R}_{n-1} + \mathbb{1}_{\{X(t_n^+)>X(t_n)\}})e^{-\beta(t_n - t_{n-1})}$$

for $R_0 = 0$ and $t_0 = 0$.

Third, set $f(\kappa_n|t_n)$ as follows [21]

$$\begin{aligned} f(\kappa_n|t_n) = &\sum_{i,j=1,j\neq i}^{h} \frac{\mathbb{1}_{\{X(t_n)=i \wedge X(t_n^+)=j\}}}{\lambda(t_n)} \\ &\times \left(q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{\boldsymbol{G}_i} \sum_{\tau\in\boldsymbol{\tau}(t_n)} \alpha e^{-\beta(t_n-\tau)}\right) \end{aligned}$$

with $\alpha\mathcal{R}$ evenly added to the conditional downgrade probability parameters q_{ij} for any $i, j \in \boldsymbol{S}$ such that $j > i$ and thus $\boldsymbol{G}_i = \sum_{j>i} \mathbb{1}_{\{q_{ij}>0\}}$ satisfying the conservation of probability and implying the loglikelihood,

$l(\boldsymbol{Q},\alpha,\beta|\boldsymbol{X})$, is defined by [21] (5.2.2.3)

$$
\begin{aligned}
l(\boldsymbol{Q},\alpha,\beta|\boldsymbol{X}) = \sum_{n=1}^{N(\bar{T})} \log\Big(&\sum_{i,j=1}^{h} \mathbb{1}_{\{X_{t_n}=i \wedge X(t_n^+)=j\}} q_{ij} \\
&+ \alpha \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{\boldsymbol{G}_i} \mathcal{R}_n \Big) \\
&- \int_0^{\bar{T}} \sum_{i=1}^{h-1} \mathbb{1}_{\{X(x)=i\}} \Big(\sum_{j=1,j\neq i}^{h} q_{ij} \Big) dx \\
&+ \sum_{\tau \in \boldsymbol{\tau}(\bar{T})} \alpha \Big(\sum_{n=1}^{N(\bar{T})} \mathbb{1}_{X(t_n^+)>X(t_n)} \\
&- \mathcal{R}_N e^{-\beta(\bar{T}-\tau_N)} \Big)
\end{aligned}
$$

for $\bar{T} = \min\{\{t : X(t) = h\} \wedge T\}$ and $N = N(\bar{T})$ as there is no contribution to $l(\boldsymbol{Q},\alpha,\beta|\boldsymbol{X})$ after a default by making a transition to state h.

Consequently, the non-Markovian model is summarized as follows for any $i, j \in \boldsymbol{S}$

- holding times, T_i,
  - in state i are IID exponential random variables with baseline intensity $q_i > 0$
  - and at a downgrade intensity $\lambda(t) > 0$ increases by an impulse of magnitude $\alpha > 0$
  - with intensity impulses accumulating and exponentially decaying with rate $\beta > 0$ over time totalling the non-Markovian intensity contribution $\alpha\mathcal{R}(t)$
- conditional transition probabilities, $\mathbb{P}(X(t+\Delta t) = j|X(t+\Delta t) \neq i, X(t) = i)$,
  - to state j given a transition out of state i are defined by $\frac{f_{i\to j}}{\lambda}$ with baseline $\frac{q_{ij}}{q_i}$
  - and $\alpha\mathcal{R}(t)$ is evenly added to conditional downgrade probability parameters q_{ij}

such that the following conditions similar to the Markovian model are satisfied for any $i, j \in \boldsymbol{S}$ [21]

- $\lambda(t) - \sum_{j=1,j\neq i} \lambda(t) f(i \to j|t) = 0$
- $0 \leq -q_{ii} = q_i = \lambda(t) - \alpha\mathcal{R}(t) \leq \lambda(t)$
- $0 \leq q_{ij} \leq q_{ij} + \frac{f(i\to j|t) - q_{ij}}{\boldsymbol{G}_i} = \lambda(t) f(i \to j|t)$ with $i \neq j$

Finally, some granularity is added to support evidence of momentum with different magnitudes and decay rates when downgraded from investment- or speculative grade ratings while maintaining a robust and parsimonious model [15] [21]. Investment grades are considered safe ratings from state 1 to $\frac{h-1}{2}$ for $h = 9$ with speculative grades being the other ratings up to the default state keeping the recursive patterns and loglikelihood function in tact and leading to an adjustment of $\mathcal{R}$ as follows [21] [46]

$$
\boldsymbol{\alpha}\mathcal{R}_n = \sum_{m=1}^{2} \sum_{\tau \in \boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m (t_n - \tau)}
$$

with $m = 1, 2$ indicating investment- or speculative grade parameter sets.

2.3 Parameter estimation

2.3.1 Markovian model

2.3.1.1 Exact maximum likelihood estimator

First, assume ratings are independent such that the MLE $\hat{\boldsymbol{Q}}$ for multiple realizations of $\boldsymbol{X}$ for any $i, j \in \boldsymbol{S}$ is defined by [21] [22] [30] [37] (5.2.3.1)

$$
\hat{q}_{ij} = \frac{N_{ij}(\bar{T})}{R_i(\bar{T})}
$$

with the number of $i \to j$ transitions, $N_{ij}(t)$, given by [21] [22] [30] [37]

$$
N_{ij}(t) = \sum_{n=1}^{N(t)} \mathbb{1}_{\{X(t_n)=i \wedge X(t_n^+)=j\}}
$$

and the summed holding times in state i, $R_i(T)$, is defined by [21] [22] [30] [37]

$$
R_i(t) = \int_0^t \mathbb{1}_{\{X(\bar{t})=i\}} d\bar{t}
$$

both among all realizations.

Second, the Hessian of the loglikelihood $\boldsymbol{H}_l(\hat{\boldsymbol{Q}}|\boldsymbol{X})$ has only nonzero diagonal entries, which are non positive as follows [21] [22] [30] [37] (5.2.3.1)

$$
(\boldsymbol{H}_l(\hat{\boldsymbol{Q}}|\boldsymbol{X}))_{kk} = \frac{\partial^2 l(\hat{\boldsymbol{Q}}|\boldsymbol{X})}{\partial q_{ij}^2} = -\frac{N_{ij}(T)}{q_{ij}^2} \leq 0
$$

with $k = h(i-1) + j$ such that for any $i_1, i_2, j_1 j_2 \in \boldsymbol{S}$ and $(i_1, j_1) \neq (i_2, j_2)$ the off-diagonal entries are given by [21] [22] [30] [37]

$$
\frac{\partial^2 l(\hat{\boldsymbol{Q}}|\boldsymbol{X})}{\partial q_{i_1 j_1} \partial q_{i_2 j_2}} = 0
$$

implying the nonzero parameter set, $\boldsymbol{V}_{\hat{\boldsymbol{Q}}}$, called the allowed pairs is strictly concave and $\hat{\boldsymbol{Q}}$ is unique. Furthermore, no other estimator $\tilde{\boldsymbol{Q}}$ with $\tilde{\boldsymbol{Q}} \neq \hat{\boldsymbol{Q}}$ attains or exceeds this global maximum (log)likelihood [54].

Finally, as $\hat{\boldsymbol{Q}}$ is diagonizable (almost surely) the corresponding $\hat{\boldsymbol{P}}$ is unique and cannot be attained by any other estimator $\tilde{\boldsymbol{Q}}$ with $\tilde{\boldsymbol{Q}} \neq \hat{\boldsymbol{Q}}$ [29] [42] [54] (5.2.3.2).

2.3.1.2 Expectation-maximization algorithm

A "classic" problem in past literature has been to obtain $\hat{\boldsymbol{Q}}$ from missing rating data. For annualized data described by a rating chain $\{X_n\}_{n\geq 0}$ the MLE for a discrete $\hat{\boldsymbol{P}}$ for any $i, j \in \boldsymbol{S}$ is defined by [30] (5.2.3.3)

$$(\hat{\boldsymbol{P}})_{ij} = p_{ij} = \frac{N_{ij}}{N_i}$$

with N_{ij} the total number of $i \to j$ transitions and N_i the number of ratings starting in state i summed over each year, however, this does not provide insight in any continuous transition probability. This problem has been optimized in recent research by the use of an EM algorithm, which allows for efficient parameter estimations of $\hat{\boldsymbol{Q}}$ with known confidence intervals, while solving both the embeddability and identification problems [21] [22] [31].

As the field of missing rating data and Markovian models have been thoroughly researched and optimized the methodologies and corresponding results are included for comparison purposes, but are not part of the scope of this research and therefore derivations have been left out [21] [22].

2.3.1.3 The delta method

It is essential to determine how confidence intervals of $\hat{\boldsymbol{Q}}$ translate to $\hat{\boldsymbol{P}}$ over time horizon $[0, T]$ for both exact MLE and the EM algorithm, which is possible by the use of the delta method. A necessary condition is a consistent MLE, which converges by definition in probability to the true $\hat{\boldsymbol{Q}}$ with asymptotic normality according to the central limit theorem (CLT) as follows [21] [38]

$$\sqrt{n}(\boldsymbol{Q} - \hat{\boldsymbol{Q}}) \xrightarrow{D} \mathcal{N}(0, \boldsymbol{\Sigma})$$

with $\boldsymbol{\Sigma}$ the covariance matrix of $\hat{\boldsymbol{Q}}$. Subsequently, there are 2 necessary conditions for consistency in $\hat{\boldsymbol{Q}}$, which are identifiability and irreducibility of the embedded rating chain $\{X_n\}_{n \leq 0} = \{X(t_n)\}_{n \leq 0}$ over time horizon $[0, T]$ with transition times $\{t_n\}_{n \leq 0}$. [49].

First, as for any realization $\boldsymbol{X}$, $\hat{\boldsymbol{Q}}$ does not allow for nonzero entries, which not driving $l(\hat{\boldsymbol{Q}}|\boldsymbol{X})$ implying that if for any $t \in [0, T]$ $\hat{\boldsymbol{P}}_1 = \hat{\boldsymbol{P}}_2$, then $\hat{\boldsymbol{Q}}_1 = \hat{\boldsymbol{Q}}_2$, such that the identifiability condition is satisfied. Additionally, as for diagonizable $\hat{\boldsymbol{Q}}$ (almost surely) the matrix exponential is a one-to-one mapping the identifiability condition is satisfied for any realization $\boldsymbol{X}$ by a contrapositive [11] [31] [38]

$$\hat{\boldsymbol{Q}}_1 \neq \hat{\boldsymbol{Q}}_2 \implies \hat{\boldsymbol{P}}_1 \neq \hat{\boldsymbol{P}}_2$$

Second, the embedded rating chain $\{X_n\}_{n \leq 0}$ is not irreducible, due to the possibility of defaulting described by absorption state, h. The definition of irreducibility is that state j is accessible from state i via some $m, n \in \mathbb{N}^+$ steps such that

$$\mathbb{P}(X_{m+n} = j | X_n = i) = \mathbb{P}(X_n = j | X_0 = i) > 0$$

for any $i, j \in \boldsymbol{S}$, which is trivially not the case for $i = h$ [53].

Finally, consistency of $\hat{\boldsymbol{Q}}$ is assumed as in recent research, however, for multiple independent realizations of $\boldsymbol{X}$ it is also claimed that irreducibility is not a necessary condition, allowing for the delta method such that the confidence intervals of $\hat{\boldsymbol{P}}$ for any $i, j \in \boldsymbol{S}$ over time horizon $[0, T]$ are defined by [2] [21] [22] [38] [49] (5.2.3.4) (5.2.3.5)

$$\begin{aligned} \text{Var}(p_{ij}(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}, t)) &\approx \Big(\frac{\partial p_{ij}(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}, t)}{\partial \boldsymbol{V}_{\hat{\hat{\boldsymbol{Q}}}}}\Big) \times (-\boldsymbol{H}(\hat{\boldsymbol{Q}})^{-1}) \\ &\times \Big(\frac{\partial p_{ij}(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}, t)}{\partial \boldsymbol{V}_{\hat{\hat{\boldsymbol{Q}}}}}\Big)^T \end{aligned}$$

with

$$p_{ij}(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}, t) = (e^{\hat{\boldsymbol{Q}}t})_{ij}$$

2.3.2 Non-Markovian model

2.3.2.1 Metropolis-Hastings algorithm

Recent research applies a Metropolis-Hastings (MH) algorithm avoiding complex first- and/or second derivatives of the (log)likelihood to obtained the posterior distribution $f_{\hat{\boldsymbol{\theta}}}(\hat{\boldsymbol{\theta}}|\boldsymbol{X})$ without computing any normalizing constant through Bayes formula as follows [21]

$$\begin{aligned} f_{\hat{\boldsymbol{\theta}}}(\hat{\boldsymbol{\theta}}|\boldsymbol{X}) &\propto L(\boldsymbol{X}|\hat{\boldsymbol{\theta}}) f_{\hat{\boldsymbol{\theta}}}(\hat{\boldsymbol{\theta}}) \\ &= L(\boldsymbol{X}|\hat{\boldsymbol{\theta}}) f_{\boldsymbol{V}_{\hat{\boldsymbol{Q}}}}(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}) f_{\hat{\boldsymbol{\alpha}}}(\hat{\boldsymbol{\alpha}}) f_{\hat{\boldsymbol{\beta}}}(\hat{\boldsymbol{\beta}}) \end{aligned}$$

with $\hat{\boldsymbol{\theta}} = \{\boldsymbol{V}_{\hat{\boldsymbol{Q}}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}\}$ a flattened matrix representation of all nonzero parameter estimators with index set $\boldsymbol{I}$, $L(\boldsymbol{X}|\hat{\boldsymbol{\theta}})$ the likelihood and $f_{\hat{\boldsymbol{\theta}}}(\hat{\boldsymbol{\theta}})$ the prior distributions.

First, $f_{\boldsymbol{V}_{\hat{\boldsymbol{Q}}}}(\boldsymbol{V}_{\hat{\boldsymbol{Q}}})$, $f_{\hat{\boldsymbol{\alpha}}}(\hat{\boldsymbol{\alpha}})$ and $f_{\hat{\boldsymbol{\beta}}}(\hat{\boldsymbol{\beta}})$ are chosen, such that they reflect prior knowledge on $\hat{\boldsymbol{\theta}}$. Following recent research, it is assumed that ratings initially have no momentum, thus $f_{\boldsymbol{V}_{\hat{\boldsymbol{Q}}}}(\boldsymbol{V}_{\hat{\boldsymbol{Q}}})$ is set equal to $\boldsymbol{V}_{\hat{\boldsymbol{Q}}}$ of the Markovian model [21].

Second, based on recent research $f_{\hat{\boldsymbol{\alpha}}}(\hat{\boldsymbol{\alpha}})$ and $f_{\hat{\boldsymbol{\beta}}}(\hat{\boldsymbol{\beta}})$ are defined by [21]

$$f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) \propto \exp(\boldsymbol{\mu}_{\hat{\boldsymbol{\alpha}}}^{-1})$$

and

$$f_{\boldsymbol{\beta}}(\boldsymbol{\beta}) \propto \exp(\boldsymbol{\mu}_{\hat{\boldsymbol{\beta}}}^{-1})$$

with appropriate means $\boldsymbol{\mu}_{\hat{\boldsymbol{\alpha}}}$ and $\boldsymbol{\mu}_{\hat{\boldsymbol{\beta}}}$ reflecting the prior knowledge that $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}} > \boldsymbol{0}$ and expected parameter values being neither near 0 or too large, which is possibly supported by CRAs pursuing stability after any rating transition and momentum being significantly present for several years after a downgrade [15] [21] [40].

Third, parameter values $\hat{\boldsymbol{\theta}}_i$ for any $i \in \boldsymbol{I}$ are drawn one-at-a-time from an unnormalized truncated normal proposal distribution with an appropriate variance $\sigma = 1e-3$ as follows [21]

$$\theta_{i,n+1} \sim \mathcal{N}(\theta_{i,n}, \sigma)$$

with $\mathbb{P}(\theta_{i,n+1}|\theta_{i,n}) = \phi(\theta_{i,n+1}|\theta_{i,n})$ the proposal distribution and $\theta_{i,n+1} \leq 0$ drawn again creating Monte Carlo Markov chains (MCMCs) $\{\theta_{i,n}\}_{n\leq 0}$. As an unnormalized truncated normal distribution remains symmetrical the acceptance probability of every draw is defined by [21]

$$\min\left\{1, \frac{\pi(\boldsymbol{X}|\theta_{i,n}, \boldsymbol{\theta}_{-i,n})\phi(\theta_{i,n+1}|\theta_{i,n})\pi(\theta_{i,n})}{\pi(\boldsymbol{X}|\theta_{i,n+1}, \boldsymbol{\theta}_{-i,n})\phi(\theta_{i,n}|\theta_{i,n+1})\pi(\theta_{i,n+1})}\right\} = \min\left\{1, \frac{\pi(\boldsymbol{X}|\theta_{i,n}, \boldsymbol{\theta}_{-i,n})\pi(\theta_{i,n})}{\pi(\boldsymbol{X}|\theta_{i,n+1}, \boldsymbol{\theta}_{-i,n})\pi(\theta_{i,n+1})}\right\}$$

with $\boldsymbol{\theta}_{-i,n}$ the first $i-1$ parameters after n draws. Drawing from a normal distribution and dismissing $\theta_{i,n+1} \leq 0$ is considered an appropriate sampling method allowing for faster computations compared to using exponential- or Gamma distributions as in recent research [21] [50].

Fourth, $\hat{\theta}_{i,N}$ is set equal to the posterior mean of the MCMCs after $N = 1e3$ iterations with $1e2$ burn-in minimizing the expected mean squared error (MSE) similar to recent research [22] [32].

Finally, sufficient conditions for convergence of the MCMCs to the unique posterior distributions of $\hat{\boldsymbol{\theta}}$ are π-irreducibility and aperiodicity [12] [51] (5.2.4.1). A sufficient condition for π-irreducibility is that for any $x, y \in \text{supp}(\phi)$ drawing y after x has a positive probability, such that $\phi(y, x) > 0$, which is trivially satisfied for $\phi \sim \mathcal{N}$ an unnormalized truncated normal distribution [51]. Next, a sufficient condition for aperiodicity is that MCMCs can remain in the same state after an iteration, such that $\mathbb{P}(\theta_{i,n+1} = \theta_{i,n}) > 0$, which is trivially true in case of rejecting a proposed parameter value [51]. The rate of convergence for MH algorithms is a classic problem on which no further research is done. Current literature does not provide a general framework to determine the rate of convergence of the MH algorithms also applicable to complex settings [57].

2.3.2.2 Projected Newton-Raphson method

First, the Newton-Raphson (NR) method is an iterative root finding method used to maximize the loglikelihood $l(\boldsymbol{\theta})$ by finding the (local) root of the partial derivatives such that $\nabla l(\boldsymbol{\theta}) = 0$. For a one-dimensional loglikelihood function $f : \mathbb{R} \to [0, 1]$ the NR method is defined by [5] (5.2.4.2)

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

implying quadratic convergence and an error propagation as follows [5] [58] (5.2.4.2)

$$\epsilon_{n+1} = -\frac{\epsilon_n^2 f'''(\xi_n)}{2f''(x_n)}$$

with ξ_n between x_n and the root x_r. An upper bound for the error ϵ_n is defined by [5] [58] (5.2.4.2)

$$\epsilon_{n+1} = \frac{1}{2}M\epsilon_n^2$$

with

$$M = \sup_{x\in\mathbb{R}} \frac{1}{2}\left|\frac{f'''(x)}{f''(x)}\right|$$

Second, the NR method can be expanded in general to a multidimensional setting of $l(\boldsymbol{\theta}|\boldsymbol{X})$, however, $\boldsymbol{D}^3 l(\boldsymbol{\theta}|\boldsymbol{X})$ with $\boldsymbol{D}$ a difference operator cannot be attained, due to the complexity $l(\boldsymbol{\theta}|\boldsymbol{X})$ and the size of the parameter space $\boldsymbol{\Theta}$ implying the upper bound for the error propagation cannot be computed [5] [58] (5.2.4.2).

Third, overall sufficient condition for the existence of a MLE is compactness of the parameter space $\boldsymbol{\Theta}$ on which $l(\boldsymbol{\theta}|\boldsymbol{X})$ is continuous [28]. Furthermore, $\hat{\boldsymbol{\theta}}$ is unique, such that every local maximum is a global maximum, if $\boldsymbol{\Theta}$ or $l(\boldsymbol{\theta}|\boldsymbol{X})$ is concave [54]. $l(\boldsymbol{\theta}|\boldsymbol{X})$ is concave in $\hat{\boldsymbol{Q}}$ and $\hat{\boldsymbol{\alpha}}$, however, not in $\hat{\boldsymbol{\beta}}$ implying that there might exist multiple stationary points for different values of $\hat{\boldsymbol{\theta}}$, which do not attain the global maximum of $l(\boldsymbol{\theta}|\boldsymbol{X})$ possibly causing incorrect convergence (5.2.4.3). Sufficient conditions for convergence of the projected NR method are as follows [5] [58]

1. $\nabla^2 l(\boldsymbol{\theta}|\boldsymbol{X}) \neq \boldsymbol{0}$ for any $\boldsymbol{\theta} > \boldsymbol{O}$
2. $\boldsymbol{D}^3 l(\boldsymbol{\theta}|\boldsymbol{X}) \neq \boldsymbol{0}$ is continuous almost everywhere on $\boldsymbol{\theta} > \boldsymbol{0}$
3. $\hat{\boldsymbol{\theta}}_0$ is sufficiently close to the root $\nabla l(\boldsymbol{\theta}_r|\boldsymbol{X}) = \boldsymbol{0}$

with risks of incorrect convergence possibly caused by [5] [58]

- Multiple stationary points
- Iteration cycles
- Poor initial guess

which can be mitigated by setting an upper bound of $2e1$ iterations, where after a new attempts is done. $\hat{\boldsymbol{\theta}}_0$ is similarly chosen according to the prior distributions used in the MH algorithm.

Fourth, to force $\hat{\boldsymbol{\theta}} > \boldsymbol{0}$, $\boldsymbol{\theta}$ is projected on $\boldsymbol{\Theta}$ as follows

$$\arg\min_{\hat{\boldsymbol{\theta}}\in\hat{\boldsymbol{\Theta}}} ||\hat{\boldsymbol{\theta}}_{i,n} - \hat{\boldsymbol{\Theta}}||$$

for any $i \in \boldsymbol{I}$ and $n \geq 0$ with $||\hat{\boldsymbol{\theta}}_{i,n} - \boldsymbol{\Theta}|| = |\hat{\boldsymbol{\theta}}_{i,n} - \{[\delta, \infty)\}|$ for some small enough $\delta = 5e - 5$. Now, $2e1$ initial guesses of $\hat{\boldsymbol{\theta}}_0$ are done to increase the probability of convergence, where after the attempt with the highest likelihood is considered as the $\hat{\boldsymbol{\theta}}$.

Finally, 2 stopping criteria are defined by the following conditions

$$\boldsymbol{\epsilon} \geq ||\hat{\boldsymbol{\theta}}_{n+1} - \hat{\boldsymbol{\theta}}_n||$$

and

$$\boldsymbol{\epsilon} \geq ||\nabla l(\hat{\boldsymbol{\theta}}_{n+1}|\boldsymbol{X}) - \nabla l(\hat{\boldsymbol{\theta}}_n|\boldsymbol{X})||$$

with $\boldsymbol{\epsilon} = 1e - 2$ such that the projected NR method should stop when close to $\boldsymbol{\theta_r}$, but continuous in case of a steep $\nabla l(\boldsymbol{\theta}|\boldsymbol{X})$ far from being optimized.

2.3.2.3 The delta method

Assuming $\hat{\boldsymbol{\theta}}$ is consistent, a closed form expression for $p_{ij}(\hat{\boldsymbol{\theta}}, t)$ is required to allow for the delta method to be applied in the non-Markovian model. Now, a closed form expression for $p_{ij}(\hat{\boldsymbol{\theta}}, t)$ over time horizon $[0, T]$ for any $i, j \in S$ considers an incountably infinite number of possible realizations from state i at time 0 to state j at time T. A path dependent closed form expression for $p_{ij}(\hat{\boldsymbol{\theta}}, T)$ is required to properly incorporate momentum as its contributions differs per realization. An attempt is to consider all possible realizations in state i and j at times 0 and T respectively with an increasing number of n transitionsover time horizon $[0, T]$. Compact notation of a path dependent closed form expression of $p_{ij}(\hat{\boldsymbol{\theta}}, T)$ might allow for an expansion to the non-Markovian model. Next, considering the Markovian model with $\boldsymbol{Q} \in \mathbb{R}^{3\times 3}$ and $q_{ij} = q \in \mathbb{R}$ for any $i, j \in S = \{1, 2, 3\}$ with $j \neq i$ the summed probability of each possible realization from state 1 at time 0 to state 3 at time T is defined by $p_{13}(T)$ as follows (5.2.4.4)

$$p_{13}(T) = \lim_{m\to\infty} \sum_{n=1}^{m} \int_0^T q^n \frac{x^{n-1}}{(n-1)!} e^{-2qx} dx$$

Unfortunately, the closed form expression for $p_{13}(T)$ according to the Markovian model indicates that applying an empirical generator matrix $\hat{\boldsymbol{Q}}$ with $q_{i_1 j_1} \neq q_{i_2 j_2}$ for any $i_1, i_2, j_2, j_2 \in \boldsymbol{S}$. The same holds for adding momentum as the probability of the embedded rating chain $\{1, 2, 3\}$ is defined as follows

$$\mathbb{P}(\{1,2,3\}) = \int_0^T \int_0^{t_2} q(q + \alpha e^{-\beta(t_2 - t_1)}) \times e^{-2(q+\alpha e^{-\beta(t_2-t_1)(t_2-t_1)})} dt_1 dt_2$$

which does not allow for a compact formulation similar to the Markovian model.

2.3.2.4 Modified thinning simulation

First, as in recent research, $\boldsymbol{P}$ over time horizon $[0, T]$ according to the non-Markovian model is empirically approximated by the use of modified thinning simulation [13] [19] [21] [46] (5.2.4.5). Modified thinning allows for independent exact simulation of the underlying exponential Hawkes processes $\boldsymbol{\lambda}(t)$ and thus of $\boldsymbol{X}(t)$ [13] [19] [21] [46] (5.2.4.5). Independent and exact simulation is used to approximate $\boldsymbol{P}$ over time horizon $[0, T]$ focusing on the PD for any initial state $i \in \boldsymbol{S}$ according the strong law of large numbers (SLLN) as follows [20] [21]

$$\mathbb{P}(\boldsymbol{X}(t) = h | \boldsymbol{X}(0) = i) = \mathbb{E}[\mathbb{1}_{\{\boldsymbol{X}(t)=h \wedge \boldsymbol{X}(0)=i\}}] = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}_{\{\boldsymbol{X}_n(t)=h \wedge \boldsymbol{X}_n(0)=i\}}$$

$\boldsymbol{P}$ is assumed to be sufficiently approximated by $1e6$ ratings.

Second, modified thinning simulation allows for the construction of confidence intervals of $\boldsymbol{P}$ via parametric bootstrapping. Assuming all available rating data resembles an entire population, then parametric bootstrapping entails point estimates of $\boldsymbol{P}$ based on subsamples drawn from the population with replacement [23]. As the point estimates of $\boldsymbol{P}$ converge to a normal distribution according to the CLT confidence intervals can be constructed [23]. Setting the size of the subsamples to the same order of magnitude as the entire population $\boldsymbol{X}$ and drawing $1e1$ subsamples might be considered sufficient to properly approximate the confidence intervals of $\hat{\boldsymbol{P}}$.

Finally, modified thinning simulation is used to generate rating data to back-test the performance of different parameter estimation methodologies in the (non-)Markovian models.

2.4 Alternative methodologies

2.4.1 Chapman-Kolmogorov equations

First, time-inhomogeneous Chapman-Kolmogorov forward, which are similar to backward-, equations are defined as follows [24]

$$\frac{\partial \boldsymbol{P}(t)}{\partial t} = \boldsymbol{P}(t)\boldsymbol{Q}(t)$$

with

$$\boldsymbol{P}(0) = \boldsymbol{I}_h$$

and might be considered to approximate $\boldsymbol{P}$ according to the non-Markovian model in combination with numerical integration [24] (5.2.5.1).

Second, an attempt is to define the expected intensity, $\mathbb{E}[\lambda_i(t)]$, for any $i \in \boldsymbol{S}$, which subsequently fully defines $\boldsymbol{Q}$. Unfortunately, this leads to complex computations as for any $i \in \boldsymbol{S}$, $\mathbb{E}[\lambda_i(t)]$ is approximated by the intensities λ_i of uncountably infinite possible realizations in state i at time t, weighted by the probability according to the non-Markovian model. Furthermore, applying $\mathbb{E}[\lambda_i(t)]$ implies a strong assumption of linearity as follows [52]

$$\mathbb{E}[f(\lambda(t))] = f(\mathbb{E}[\lambda(t)])$$

for any probability function $f : (0, \infty) \to [0, 1]$, which is trivially incorrect for the underlying exponential Hawkes process.

Third, the incorrectness of the strong assumption of linearity is ignored and the non-Markovian model is adjusted according to the strong, but reasonable assumptions that momentum disappears after an upgrade and that for any realization defaulting over time horizon $[0, T]$ will do so by consecutive downgrades referred to as the heuristic estimator. This implies that for ratings in state 1 at time 0 that $\mathbb{E}[\lambda_1(t)] = \lambda_1(t)$, which in turn fully defines $\mathbb{E}[\lambda_2(t)]$. $\mathbb{E}[\lambda_2(t)]$ is now defined as follows

$$\mathbb{E}[\lambda_2(t)] = q_2 + \left(\frac{q_{12}}{q_1}\right) \times t \times \int_0^t q_1 e^{-q_1\bar{t}} \alpha_1 e^{-\beta_1(t-\bar{t})} d\bar{t}$$

A general consequence is that $\lambda_2(t)$ is defined by the baseline intensity q_2 and non-Markovian contribution of a $1 \to 2$ transition before time t weighted by its probability. Similarly, for ratings in state 1 at time 0, $\lambda_3(t)$ is defined as follows

$$\begin{aligned}\mathbb{E}[\lambda_3(t)] = q_3 \\ &+ \frac{q_{13}}{q_1} \times t \times \int_0^t q_1 e^{-q_1\bar{t}} \alpha_1 e^{-\beta_1(t-\bar{t})} d\bar{t} \\ &+ \frac{q_{23} + \frac{\gamma_2(t)}{N_2}}{q_3 + \gamma_2} \times t \\ &\times \int_0^t (q_2 + \gamma_2(\bar{t})) e^{-(q_2+\gamma_2(\bar{t}))\bar{t}} \alpha_1 e^{-\beta_1(t-\bar{t})} d\bar{t}\end{aligned}$$

with

$$\gamma_i(t) = \mathbb{E}[\lambda_i(t)] - q_i$$

such that in general for any $i \in \boldsymbol{S}$ with $\gamma_1(t) = 0$, $\mathbb{E}[\lambda_i(t)]$ is given by

$$\begin{aligned}\mathbb{E}[\lambda_j] = \sum_{i=1}^{j} \frac{q_{ij} + \frac{\gamma_i}{N_i}}{q_i + \gamma_i(t)} \\ &\times \int_0^t (q_i + \gamma_i(\bar{t})) e^{-(q_i+\gamma_i(\bar{t}))\bar{t}} \\ &\times \alpha_m e^{-\beta_m(t-\bar{t})} d\bar{t}\end{aligned}$$

for $m = \mathbb{1}_{\{i>4\}} + 1$.

2.4.2 Discretized simulation

To overcome the computational requirements for computing $\boldsymbol{P}$ with confidence intervals via modified thinning simulation an attempt is made to use linear approximations with small enough discretized timesteps Δt similar to the Markovian model as follows

$$\boldsymbol{P}(t_1, t_2) \approx (\mathbf{I} + \boldsymbol{Q}\frac{t_2 - t_1}{n})^n$$

for some $t_1, t_2 \in [0, T]$ with $n\Delta t = t_2 - t_1$ and $n \in \mathbb{N}^+$. Unfortunately, this this method cannot directly be applied to the non-Markovian model, which is demonstrated for the embedded rating chains $\{1, 1, 1, 3\}$ and $\{1, 2, 1, 3\}$ as follows

$$\mathbb{P}(\{1, 1, 1, 3\}) = (1 - q\Delta t) \times (1 - q\Delta t) \times (q\Delta t)$$

and

$$\mathbb{P}(\{1, 2, 1, 3\}) = (q\Delta t) \times (q\Delta t) \times ((q + \alpha e^{-\beta(2\Delta t)})\Delta t)$$

with $q_{ij} = q$ for any $i, j \in S$, implying $(\boldsymbol{Q})_{13}$ should take on the 2 different values q and $q + \alpha e^{-\beta(2\Delta t)}$ to approximate the probability of both embedded rating chains correctly.

Now, an attempt is to create an iterative algorithm properly incorporating momentum for $m = 1$, thus not differentiating between investment- or speculative grade ratings, with $\boldsymbol{Q} \in \mathbb{R}^{3\times3}$, $\alpha, \beta > 0$, $T \in \mathbb{R}^+$ and $N \in \mathbb{N}^+$ defined by pseudo-code as follows

Input: $\boldsymbol{Q}, \alpha, \beta, T, N$

1 Initialize $\boldsymbol{\Psi}_1 = \mathbf{0} \in \mathbb{R}^{3\times3}$;
$\boldsymbol{\Phi}_1 = \mathbf{0} \in \mathbb{R}^{3\times3}$;
$\boldsymbol{\Psi}_2 = \hat{\boldsymbol{Q}}_{1,d} \cdot e^{-\beta\Delta t} \in \mathbb{R}^{3\times3}$;
$\boldsymbol{\Phi}_2 = \hat{\boldsymbol{Q}}_{1,d} \cdot e^{-\beta\Delta t} \otimes \boldsymbol{N}_{\alpha,\Delta t} \in \mathbb{R}^{3\times3}$;

2 Set $\Delta t = T/N$

3 **for** $n = 3, ...N$

5 Set $\boldsymbol{Q}_n = \boldsymbol{Q}_{n-1} \times (\hat{\boldsymbol{Q}} e^{-\beta\Delta t} + \hat{\boldsymbol{Q}}_d)$

4 Set $\boldsymbol{\Phi}_n = \boldsymbol{Q}_n \otimes \boldsymbol{N}_{\alpha,\Delta t} +$
$(\hat{\boldsymbol{Q}}_{1,u} \times \hat{\boldsymbol{Q}}_u^{n-3} \times \hat{\boldsymbol{Q}}_d) \cdot e^{-\beta\Delta t} \otimes \boldsymbol{N}_{\alpha,\Delta t} +$
$\boldsymbol{\Phi}_{n-1} \times \hat{\boldsymbol{Q}} +$
$\boldsymbol{\Phi}_{n-1} \cdot e^{-(n-1)\beta\Delta t} \otimes N_{\alpha,\Delta t}$

for

$$\hat{\boldsymbol{Q}} = \begin{pmatrix} 1 - q_1\Delta t & q_{12}\Delta t & q_{13}\Delta t \\ q_{21}\Delta t & 1 - q_2\Delta t & q_{23}\Delta t \\ 0 & 0 & 0 \end{pmatrix}$$

with $\hat{\boldsymbol{Q}}_u$ the upper triangular matrix of $\hat{\boldsymbol{Q}}$ including the main diagonal resembling non-downgrade transitions and $\hat{\boldsymbol{Q}}_d$ the lower triangular matrix of $\hat{\boldsymbol{Q}}$ excluding the main diagonal resembling downgrades. $\hat{\boldsymbol{Q}}_{1,u}$ and $\hat{\boldsymbol{Q}}_{1,d}$ are constructed similarly, but only include nonzero entries on the first row describing the initial state 1 and for any $\boldsymbol{M} \in \mathbb{R}^{3\times3}$ (5.2.5.2)

$$\boldsymbol{M} \otimes \boldsymbol{N}_{\alpha,\Delta t} = \begin{pmatrix} -\Upsilon_1 & \frac{\Upsilon_1}{2} - \Upsilon_2 & \frac{\Upsilon_1}{2} + \Upsilon_2 \\ 0 & \frac{\Upsilon_1}{2} - \Upsilon_2 & \frac{\Upsilon_1}{2} + \Upsilon_2 \\ 0 & 0 & 0 \end{pmatrix}$$

with

$$\Upsilon_j = \alpha\Delta t \sum_{i=1}^{3} (\boldsymbol{M})_{ij}$$

for any $j \in \{1, 2\}$ with $(\boldsymbol{P}(n \cdot \Delta t))_{13} = ((\mathbf{I}_3 + \boldsymbol{Q}\Delta t)^n)_{13} + (\sum_i^n \boldsymbol{\Phi}_i)_{13}$ for some $\boldsymbol{\Phi}_n \in \mathbb{R}^{3\times3}$.

The rationale behind the algorithm is that $\boldsymbol{\Phi}$ incorporates non-Markovian intensity contributions for every possible embedded rating chain weighted by its probability, however, many complex cross products have to be considered for properly simulating the non-Markovian model in this manner. For $n = 1, 2, 3, 4$, $\boldsymbol{\Phi}_n$ is demonstrated by writing out the non-Markovian intensity contributions for all possible embedded rating chains weighted by their probability (5.2.5.2). Unfortunately, this iterative algorithm does not properly incorporate the momentum, therefore not correctly approximate $\boldsymbol{P}$ for small enough timestep Δt.

2.4.3 Modified Markovian model

The modified Markovian model over time horizon $[0, T]$ is defined by the Markovian model applied to realizations with the same initial state such that $\boldsymbol{X}^i = \{X(t) : X(0) = i \wedge t \in [0,T]\}$ for any $i \in \boldsymbol{S}$. Subsequently, using exact MLE and the delta method the corresponding $\hat{\boldsymbol{Q}}^i$ and modified $\hat{\boldsymbol{P}}^i$ are approximated with confidence intervals for any $i \in \boldsymbol{S}$. It is hypothesized that the modified Markovian model might sufficiently incorporate momentum and result into a better approximation of $\hat{\boldsymbol{P}}$ compared to the Markovian model. The rationale behind this model is that, in short, separating realizations based on their initial state $i \in \boldsymbol{S}$ as for $\boldsymbol{X}_i$ minimizes the intertwining (non-)Markovian behaviour.

The intertwining of (non-)Markovian behaviour can be demonstrated by only considering realizations with initial state, e.g., 1, such that $\boldsymbol{X}^1 = \{X(t) : X(0) = 1 \wedge t \in [0,T]\}$. Similar to $\hat{\boldsymbol{Q}} \in \mathbb{R}^{9\times 9}$ according to the Markovian model, the nonzero entry, by assumption, q^1_{89} of $\hat{\boldsymbol{Q}}^1$ is driven by the total number of $8 \to 9$ transitions, $N^1_{89}(T)$, and summed holding times $\boldsymbol{R}^1_8(T)$ in state 8 among all realizations over time horizon $[0,T]$ [21] [37] [30] [22]. Any realization $\boldsymbol{X} \in \boldsymbol{X}^1$ making the $8 \to 9$ transition over time horizon $[0,T]$ it most likely has been downgraded several times and at least once, based on common empirical generator matrices $\hat{\boldsymbol{Q}}$ [21] [37] [30] [22]. If all realizations $\boldsymbol{X} \in \boldsymbol{X}^1$ making the $8 \to 9$ transition somewhere over time horizon $[0,T]$ have similar momentum, then $\hat{q}^1_{89}$ is solely based on realizations with comparable stochastic behaviour. Next, consider $\hat{q}_{89}$ based on realizations with initial states 1 or 8 such that $\boldsymbol{X} \in \boldsymbol{X}^1 \cup \boldsymbol{X}^8$. Now, $\hat{q}_{89}$ is again defined by the number of $8 \to 9$ transitions, $N_{89}(T)$, and the summed holding times in state 8, $R_8(T)$, among all ratings over time horizon $[0,T]$ which are driven by realizations with most likely significant momentum starting in state 1 or none starting in state 8. Reducing this spread in stochastic behaviour might improve the performance of approximating $\hat{\boldsymbol{P}}$ incorporating momentum significantly using a straightforward (modified) Markovian model allowing for confidence intervals.

The ability of the modified Markovian model to properly incorporate momentum can be mathematically demonstrated. Considering realizations $\boldsymbol{X} \in \boldsymbol{X}^1$ making the $8 \to 9$ transition somewhere over time horizon $[0,T]$. Let $t_1, t_2 \in [0,T]$ with $t_1 < t_2$ and $\Delta t = t_2 - t_1$ and make the reasonable assumption that all realizations $\boldsymbol{X} \in \boldsymbol{X}^1$ in state 8 at time t_1 have underlying intensity similar to $\hat{\lambda}(t)$. This assumption can be considered reasonable as any realization $\boldsymbol{X} \in \boldsymbol{X}^1$ most likely has significant accumulated momentum when making a transition to state 8 driving shorter holding times with lower probability that momentum has decayed before leaving state 8 again. Also, ratings are empirically most likely to get downgraded 1 or 2 states at once implying that the accumulated momentum for any realization $\boldsymbol{X} \in \boldsymbol{X}^1$ in state 8 is indeed comparable [21] [37] [30] [22]. Now, by setting

$$q_8 = \int_{t_1}^{t_2} \hat{\lambda}(t)dt$$

and given $\hat{\lambda}(t)$

$$q_{89} = f(8 \to 9|\hat{\lambda}(t_1), t_1)$$

then $\mathbb{P}(X(t_2) = 9|X(t_1) = 8)$ is approximated for fixed t_2 by $e^{-q_8\Delta t}q_{89}$ for any $\boldsymbol{X} \in \boldsymbol{X}^1$. As realizations $\boldsymbol{X} \in \boldsymbol{X}^1$ most likely have significant momentum in state 8, the holding times T_8 are expected to be relatively short. Relative short holding times, T_8, imply a relative small spread as the variance is inversely proportional to $\hat{\lambda}(t)$ such that q_8 and q_{89} become even more reasonable approximations. In general, the higher the expected momentum and thus $\hat{\lambda}(t_1)$ the lower the variance among T_i for any $i \in \boldsymbol{S}$, which allows the modified Markovian model to be an appropriate estimator at any time $t \in [0,T]$. If the expected momentum $\hat{\lambda}(t)$ for any state $i \in S$ at any time $t \in [0,T]$ is low, then a (modified) Markovian model is an appropriate estimator in the first place.

Finally, the first disadvantage of the modified Markovian model is that the $\boldsymbol{X}^i$ for any $i \in \boldsymbol{S}$ is a subset of $\boldsymbol{X}$ and thus smaller implying larger confidence intervals of $\hat{\boldsymbol{P}}$. Furthermore, a second disadvantage is that all realizations $X \in \boldsymbol{X}^i$ for any $i \in \boldsymbol{S}$ have no momentum by assumption, which might be considered a strong assumption and decreases the size of of the rating data set even further by disallowing realizations recently downgraded.

3 Results

3.1 Modified thinning simulation

The input for the modified thinning simulation is chosen similar to recent research with $h = 9$ and thus $\boldsymbol{S} = \{1, ..., 9\}$ with initial number of ratings per state as follows

$$413 \quad 1313 \quad 2232 \quad 2318 \quad 2021 \quad 4504 \quad 1333 \quad 59$$

such that corresponding labels of the ratings are defiend by [21]

$$Aaa \quad Aa \quad A \quad Baa \quad Ba \quad B \quad Caa \quad Ca \quad D$$

Non-Markovian momentum parameters $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^{1\times 2}$ are given by

$$\boldsymbol{\alpha} = \begin{pmatrix} 3.10 & 12.91 \end{pmatrix} \times 10^{-2}$$
$$\boldsymbol{\beta} = \begin{pmatrix} 352.34 & 170.95 \end{pmatrix} \times 10^{-2}$$

and the generator matrix $\boldsymbol{Q} \in \mathbb{R}^{9\times 9}$ is as follows [21]

$$\boldsymbol{Q} = \begin{pmatrix} -8.69 & 8.36 & 0.31 & 0 & 0.02 & 0 & 0 & 0 & 0 \\ 1.17 & -10.88 & 9.42 & 0.25 & 0.03 & 0.01 & 0 & 0 & 0 \\ 0.06 & 2.40 & -9.38 & 6.66 & 0.17 & 0.07 & 0.02 & 0 & 0 \\ 0.02 & 0.16 & 3.87 & -9.47 & 4.96 & 0.40 & 0.06 & 0 & 0 \\ 0.01 & 0.06 & 0.33 & 6.36 & -17.74 & 10.60 & 0.37 & 0.01 & 0 \\ 0 & 0.03 & 0.12 & 0.35 & 5.03 & -16.10 & 10.12 & 0.40 & 0.04 \\ 0 & 0.02 & 0.01 & 0.13 & 0.48 & 10.28 & -19.76 & 6.22 & 2.61 \\ 0 & 0 & 0.18 & 0.29 & 0.50 & 4.47 & 13.46 & -28.38 & 9.48 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \times 10^{-2}$$

resulting in the independent and exact simulation of 14,193 realizations according to the non-Markovian model over time horizon $[0, T]$ with $T = 30$ using modified thinning as is demonstrated in figure 1.

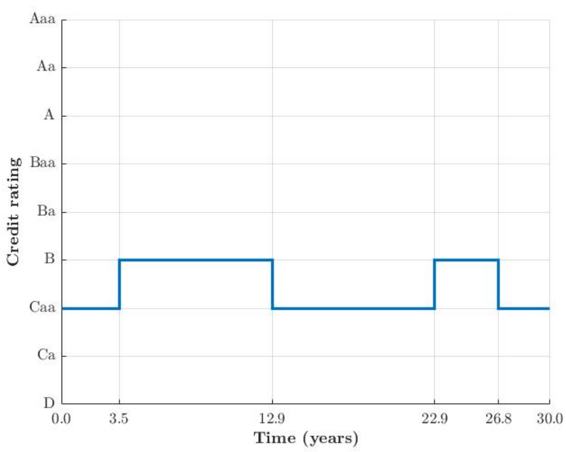

Figure 1: Simulation of randomly chosen realization according to the non-Markovian model over time horizon $[0, T]$ using modified thinning with $h = 9$ and $T = 30$.

To confirm that the momentum is properly incorporated by the modified thinning simulation, T_i can be compared to the expected truncated exponential distribution for any $i \in \boldsymbol{S}$ according to the Markovian model solely based on $\boldsymbol{Q}$ as is shown in figure 2.

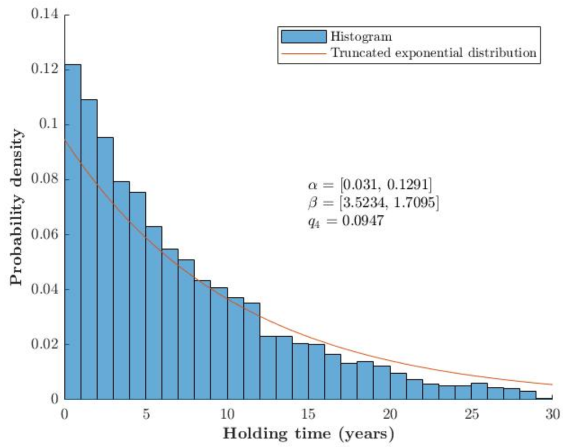

Figure 2: Distribution of T_4 according to the non-Markovian model with expected truncated exponential distribution solely based on $q_4 > 0$ according to the Markovian model.

In figure 2, T_4 is shifted towards shorter holding times compared to the expected truncated exponential distribution with $q_4 > 0$ according to the Markovian model. Momentum drives a decrease in holding times, therefore figure 2 is according to the expected behaviour of the non-Markovian model.

The modified thinning simulation algorithm with rejected or accepted (thinned) proposed holding times based on the intensity $\lambda(t)$, local maximum intensity $\bar{\lambda}(t)$ and $u \sim \mathcal{U}(0, 1)$ is demonstrated in figure 3.

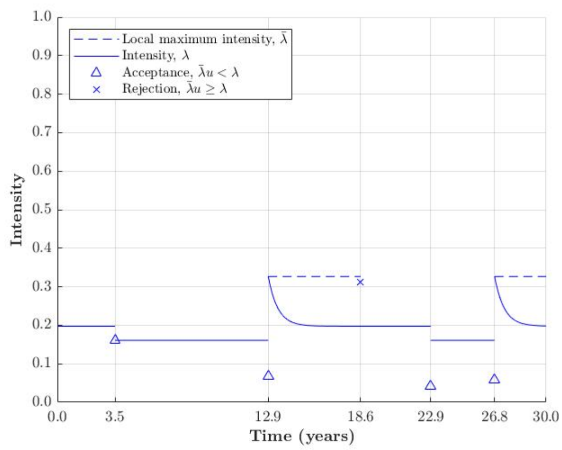

Figure 3: Simulation of $\lambda(t)$ for randomly chosen realization (corresponding to figure 1) with demonstrated rejection or acceptance of proposed holding times based on $\lambda(t)$, local maximum intensity $\bar{\lambda}(t)$ and $u \sim \mathcal{U}(0, 1)$ according to the non-Markovian model over time horizon $[0, T]$ using modified thinning.

The duration of modified thinning simulation according to the Markovian model for the described setting is ~ 5 seconds.

3.2 Markovian model

3.2.1 Exact maximum likelihood estimator

By means of the exact MLE according to the Markovian model the following generator matrix $\hat{\boldsymbol{Q}}_1 \in \mathbb{R}^{9\times 9}$ is computed

$$\hat{\boldsymbol{Q}}_1 = \begin{pmatrix} -8.55 & 8.25 & 0.30 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.15 & -10.97 & 9.50 & 0.28 & 0.03 & 0.02 & 0 & 0 & 0 \\ 0.05 & 2.36 & -9.38 & 6.70 & 0.18 & 0.07 & 0.03 & 0 & 0 \\ 0.01 & 0.14 & 3.89 & -9.36 & 4.87 & 0.39 & 0.07 & 0 & 0 \\ 0.01 & 0.08 & 0.34 & 6.42 & -18.06 & 10.81 & 0.37 & 0.03 & 0 \\ 0 & 0.02 & 0.11 & 0.37 & 5.04 & -16.63 & 10.37 & 0.54 & 0.19 \\ 0 & 0.01 & 0 & 0.15 & 0.47 & 10.46 & -21.43 & 6.97 & 3.36 \\ 0 & 0 & 0.20 & 0.28 & 0.55 & 4.34 & 13.39 & -30.42 & 11.67 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \times 10^{-2}$$

which in itself does not provide any insight as the momentum parameters are left out, however, $\hat{\boldsymbol{P}}_1(t) = e^{\hat{\boldsymbol{Q}}_1 t}$ can be compared to a sufficient approximation of $\boldsymbol{P}$ using modified thinning simulation with $1e6$ realizations. This comparison is made together with $\hat{\boldsymbol{P}}_2$, $\hat{\boldsymbol{P}}_3$, $\hat{\boldsymbol{P}}_4$, $\hat{\boldsymbol{P}}_5$ and $\hat{\boldsymbol{P}}_6$ according to the EM algorithm, modified Markovian model and modified thinning simulation following parameter estimations via the MH algorithm and projected NR method in figure 9.

The duration of exact maximum likelihood estimator according to the Markovian model for the described setting is ~ 1 minute.

3.2.2 Expectation-maximization algorithm

The duration of the EM algorithm according to the Markovian model is ~ 3 minutes for the described setting with the stopping criteria $\epsilon = 1e-9$ chosen similar to recent research and is applied to missing data [48] [21] [22].

By means of the MLE via the EM algorithm according to the Markovian model the following generator matrix $\hat{\boldsymbol{Q}}_2 \in \mathbb{R}^{9\times 9}$ is computed

$$\hat{\boldsymbol{Q}}_2 = \begin{pmatrix} -8.43 & 8.16 & 0.27 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.13 & -11.04 & 9.53 & 0.28 & 0.05 & 0.05 & 0 & 0 & 0 \\ 0.04 & 2.39 & -9.38 & 6.67 & 0.19 & 0.05 & 0.02 & 0 & 0 \\ 0.01 & 0.14 & 3.86 & -9.33 & 4.83 & 0.40 & 0.06 & 0 & 0 \\ 0.01 & 0.08 & 0.39 & 6.39 & -18.03 & 10.45 & 0.52 & 0.18 & 0 \\ 0 & 0.02 & 0.11 & 0.36 & 5.05 & -16.39 & 9.85 & 0.56 & 0.45 \\ 0 & 0.02 & 0 & 0.17 & 0.44 & 10.46 & -20.80 & 6.52 & 3.19 \\ 0 & 08 & 0.20 & 0.25 & 0.60 & 4.18 & 13.66 & -29.33 & 10.44 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \times 10^{-2}$$

which again does not provide any insight, however, $\hat{\boldsymbol{P}}_2(t) = e^{\hat{\boldsymbol{Q}}_2 t}$ is compared to a sufficient approximation of $\boldsymbol{P}$ using modified thinning simulation with $1e6$ realizations in figure 9. It is noticeable that $\hat{\boldsymbol{Q}}_1$ and $\hat{\boldsymbol{Q}}_2$ are similar with the largest deviation in parameter entries $< 1.1e-2$ therefore expecting to translate in similar $\hat{\boldsymbol{P}}_1$ and $\hat{\boldsymbol{P}}_2$.

3.3 Non-Markovian model

3.3.1 Metropolis-Hastings algorithm

The MH algorithm is demonstrated in figure 4 by plotting the MCMCs of $\hat{\boldsymbol{\alpha}}_3$ and $\hat{\boldsymbol{\beta}}_3$ with corresponding posterior distribution, posterior mean and real modified thinning simulation input. In figure 12 and 13 (5.1) the MH algorithm is demonstrated similarly for $\hat{\boldsymbol{Q}}_3$ (5.1).

The duration of the MH algorithm according to the non-Markovian model is ~ 6 hours for the described setting with most importantly $1e3$ iterations, $1e2$ burn-in, $\boldsymbol{\mu}_{\hat{\boldsymbol{\alpha}}} = [0.1;\ 1]$ and $\boldsymbol{\mu}_{\hat{\boldsymbol{\beta}}} = [10;\ 1]$.

By setting $\hat{\boldsymbol{\theta}}_3$ equal to the posterior means of $\{\theta_n\}_{n\geq 0}$ the MSE is minimized after applying the MH algorithm according to the non-Markovian model. The following non-Markovian momentum parameters $\hat{\boldsymbol{\alpha}}_3, \hat{\boldsymbol{\beta}}_3 \in \mathbb{R}^{1\times 2}$ are computed

$$\hat{\boldsymbol{\alpha}}_3 = \begin{pmatrix} 1.95 & 12.26 \end{pmatrix} \times 10^{-2}$$
$$\hat{\boldsymbol{\beta}}_3 = \begin{pmatrix} 338.84 & 181.06 \end{pmatrix} \times 10^{-2}$$

and the generator matrix $\hat{\boldsymbol{Q}}_3 \in \mathbb{R}^{9\times 9}$ is computed

$$\hat{\boldsymbol{Q}}_3 = \begin{pmatrix} -8.49 & 8.17 & 0.32 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.15 & -10.95 & 9.48 & 0.28 & 0.03 & 0.02 & 0 & 0 & 0 \\ 0.05 & 2.36 & -9.35 & 6.68 & 0.17 & 0.06 & 0.02 & 0 & 0 \\ 0.01 & 0.14 & 3.89 & -9.35 & 4.87 & 0.38 & 0.07 & 0 & 0 \\ 0.01 & 0.08 & 0.34 & 6.43 & -18.01 & 10.79 & 0.34 & 0.18 & 0 \\ 0 & 0.02 & 0.11 & 0.37 & 5.03 & -16.19 & 10.23 & 0.39 & 0.04 \\ 0 & 0.02 & 0.01 & 0.15 & 0.47 & 10.44 & -20.23 & 6.42 & 2.71 \\ 0 & 0 & 0.21 & 0.28 & 0.55 & 4.33 & 13.40 & -28.39 & 9.61 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \times 10^{-2}$$

3.3.2 Projected Newton-Raphson method

By means of the MLE via the projected NR method according to the non-Marovian model the following generator matrix $\hat{\boldsymbol{Q}}_4 \in \mathbb{R}^{9\times 9}$ is computed

$$\hat{\boldsymbol{Q}}_4 = \begin{pmatrix} -8.55 & 8.25 & 0.30 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.15 & -10.97 & 9.50 & 0.28 & 0.02 & 0.02 & 0 & 0 & 0 \\ 0.05 & 2.36 & -9.35 & 6.69 & 0.17 & 0.06 & 0.02 & 0 & 0 \\ 0.01 & 0.14 & 3.89 & -9.33 & 4.86 & 0.38 & 0.06 & 0 & 0 \\ 0.01 & 0.08 & 0.34 & 6.42 & -17.99 & 10.79 & 0.34 & 0.01 & 0 \\ 0 & 0.02 & 0.11 & 0.37 & 5.04 & -16.19 & 10.22 & 0.39 & 0.04 \\ 0 & 0.01 & 0.01 & 0.15 & 0.47 & 10.46 & -20.27 & 6.43 & 2.73 \\ 0 & 0 & 0.20 & 0.28 & 0.55 & 4.34 & 13.39 & -28.36 & 9.60 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \times 10^{-2}$$

and non-Markovian momentum parameters $\hat{\boldsymbol{\alpha}}_4, \hat{\boldsymbol{\beta}}_4 \in \mathbb{R}^{1\times 2}$ are computed

$$\hat{\boldsymbol{\alpha}}_4 = \begin{pmatrix} 2.27 & 13.48 \end{pmatrix} \times 10^{-2}$$
$$\hat{\boldsymbol{\beta}}_4 = \begin{pmatrix} 340.56 & 205.37 \end{pmatrix} \times 10^{-2}$$

after 9 iterations by the selected attempt leading to the MLE, however, this attempt by the projected NR method already seemed to have converged sufficiently after approximately 5 iterations as demonstrated in table 1 and figure 5. The second stopping criteria $\boldsymbol{\epsilon} \geq \nabla l(\boldsymbol{\theta}_{n+1}|\boldsymbol{X}) - \nabla l(\boldsymbol{\theta}_n|\boldsymbol{X})$ might not be satisfied after 5 iterations if the absolute partial derivative $|\nabla l(\boldsymbol{\theta}_n|\boldsymbol{X})| > 1$ in some dimension $i \in \boldsymbol{I}$, implying $\nabla l(\boldsymbol{\theta}_{n+1}|\boldsymbol{X}) - \nabla l(\boldsymbol{\theta}_n|\boldsymbol{X})$ will decrease slower than $|\boldsymbol{\theta}_{n+1} - \boldsymbol{\theta}_n|$ demanding more iterations even if near $\boldsymbol{\theta}_r$ already. The projected NR method is demonstrated in figure 6 by plotting the iterative values of $\hat{\boldsymbol{\alpha}}_4, \hat{\boldsymbol{\beta}}_4 \in \mathbb{R}^{1\times 2}$.

The duration of the projected NR method according to the non-Markovian model is ~ 1 hour for the described setting with $1e2$ initial guesses for $\hat{\boldsymbol{\theta}}_{4,0}$ with a maximum number of iterations of $1e2$, a stopping criteria of $\epsilon = 1e-2$ and $\delta = 5e-5$.

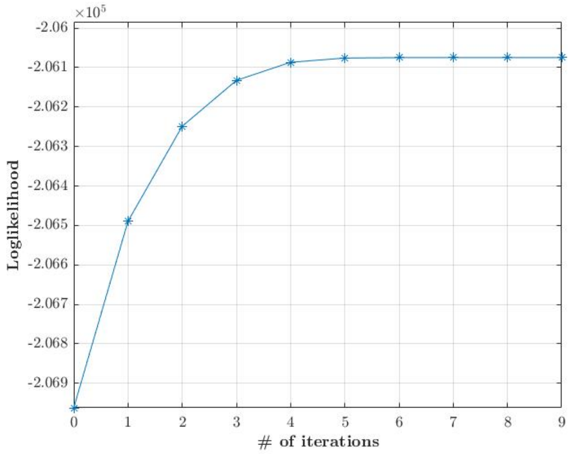

Figure 5: $l(\hat{\boldsymbol{\theta}}|\boldsymbol{X})$ of selected attempt during projected NR method with $1e2$ initial guesses for $\hat{\boldsymbol{\theta}}_{4,0}$ with a maximum number of iterations of $1e2$, a stopping criteria of $\boldsymbol{\epsilon} = 1e-2$ and $\delta = 5e-5$.

First, in figure 6 clear convergence to the neighbourhood of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is shown, however, not precisely. Precision might be influenced by size of the rating data set, which might be supported by improved estimations of $\boldsymbol{P}$ by $\hat{\boldsymbol{P}}_4$ using datasets of different sizes in figure 11 (5.1). Adding granularity to the non-Markovian model by making a distinction between investment- and speculative grade might ratings decrease the precision as there is relatively less rating data per estimated in $\hat{\boldsymbol{\theta}}_4$.

Second, in figure 6, the iterative values of $\hat{\boldsymbol{\alpha}}_4$ and $\hat{\boldsymbol{\beta}}_4$ do not monotonically converge to the roots $\hat{\boldsymbol{\alpha}}_r, \hat{\boldsymbol{\beta}}_r \in \mathbb{R}_r^{1\times 2}$ which should be the case for a fully concave loglikelihood function. This behaviour might be caused by a complex $l(\hat{\boldsymbol{\theta}}_4|\boldsymbol{X})$ with multiple roots.

Third, in figure 7, 90% confidence intervals of $\hat{\boldsymbol{P}}_4$ are shown for $1e1$ parameter estimations from subsamples of $1e4$ realizations via parametric bootstrapping and sufficiently approximated by $1e6$ ratings. One clear outlier might be caused by the projected NR method converging to a stationary point not equal to the global maximum, therefore also demonstrating the weakness of this parameter estimation methodology.

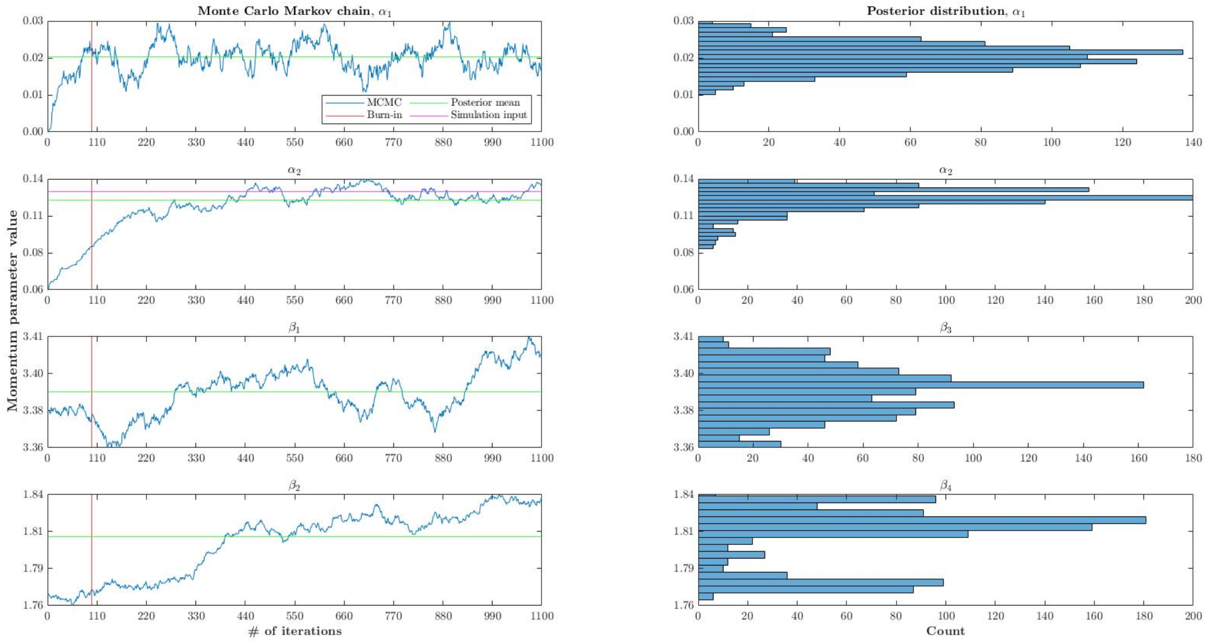

Figure 4: MCMCs and corresponding posterior distribution, posterior mean and modified thinning simulation input of $\hat{\boldsymbol{\alpha}}_3, \hat{\boldsymbol{\beta}}_3$ during the MH algorithm for $1e3$ iterations with $1e2$ burn-in.

Table 1: $\hat{\boldsymbol{\alpha}}_{4,0}, \hat{\boldsymbol{\beta}}_{4,0}$, $\hat{\boldsymbol{\alpha}}_4, \hat{\boldsymbol{\beta}}_4$, $l(\boldsymbol{\theta}|\boldsymbol{X})$ and number of step before satisfying the stopping criteria with $\boldsymbol{\epsilon} = 1e-2$ or attaining 20 iterations during $1e2$ initial guesses for $\boldsymbol{\theta}_0$ during the projected NR method according to the non-Markovian model with $1e2$ initial guesses for $\hat{\boldsymbol{\theta}}_{4,0}$, a maximum number of iterations of $1e2$, a stopping criteria of $\boldsymbol{\epsilon} = 1e-2$ and $\delta = 5e-5$..

$\hat{\boldsymbol{\alpha}}_0, \hat{\boldsymbol{\beta}}_0$ $\times 1e-2$	$\hat{\boldsymbol{\alpha}}_4, \hat{\boldsymbol{\beta}}_4$ $\times 1e-2$	$l(\boldsymbol{\theta}\|\boldsymbol{X})$ $\times 1e3$	Steps
$[9.13, 3.92]$ $[462.48, 147.15]$	$[17.80e6, 0.01]$ $[10.30e8, 0.01]$	-206.35	20
$[28.23, 9.41]$ $[1255.80, 115.60]$	$[0.01, 0.01]$ $[5.89e4, 0.01]$	-206.70	20
$[3.72, 4.58]$ $[143.03, 105.61]$	$[93.38e5, 0.01]$ $[36.30e7, 0.01]$	-206.45	20
$[1.11, 5.24]$ $[607.87, 351.41]$	$[9.00, 0.01]$ $[52.50e7, 0.11]$	-206.69	20
$[0.01, 22.79]$ $[1206.21.91, 98.75]$	$[0.01, 47.03e6]$ $[0.06, 21.91e7]$	-210.57	20
$[0.39, 5.19]$ $[584.34, 170.47]$	$[0.01, 13.51]$ $[62.29e3, 205.63]$	-206.13	13
$[2.40, 0.24]$ $[653.70, 26.68]$	$[0.01, 0.01]$ $[58.93e3, 0.01]$	-206.70	14
$[1.36, 12.62]$ $[583, 43, 144, 97]$	$[0.01, 13, 50]$ $[0.01, 205, 50]$	-206.13	20
$[2.26, 27.31]$ $[186.20, 154.10]$	$[37.24e3, 0.01]$ $[43.88e4, 0.01]$	-207.75	20
$[2.96, 6.97]$ $[485.91, 211.36]$	$[20.91e5, 21.73e6]$ $[77.72e5, 100.83]$	-210.92	20
$[1.50, 17.13]$ $[68.05, 2.99]$	$[0.01, 6.30]$ $[58.87e3, 35.14e3]$	-206.69	20
$[0.97, 13.78]$ $[189.47, 380.84]$	$[1.23, 0.01]$ $[15.58e7, 0.01]$	-206.70	20
$[0.30, 1.66]$ $[70.89, 229.80]$	$[2.16, 0.01]$ $[280.00, 11.14e4]$	-206.63	16
$[1.61, 48.98]$ $[301.27, 198.72]$	$[0.01, 83.16e6]$ $[2.77, 36.27e7]$	-210.07	20
$[5.65, 22.23]$ $[303.24, 403.79]$	$[10.87, 0.01]$ $[78.90e7, 0.01]$	-206.87	20
$[1.59, 9.72]$ $[127.72, 61.00]$	$[46.61e3, 0.01]$ $[129.55, 11.12]$	-206.14	20
$[1, 39, 43, 81]$ $[943.02, 195.00]$	$[0.01, 79.04e6]$ $[53.71e7, 37.75e7]$	-210.05	20
$[1.96, 5.47]$ $[316.20, 33.93]$	$[2.27, 13.48]$ $[340.56, 205.37]$	-206.07	9
$[1.03, 0.41]$ $[222.81, 192.06]$	$[2.16, 0.01]$ $[276.57, 11.14e4]$	-206.63	15
$[6.97, 6.36]$ $[88.08, 146.90]$	$[15.64e7, 0.01]$ $[11.12e6, 13.50]$	-207.02	20

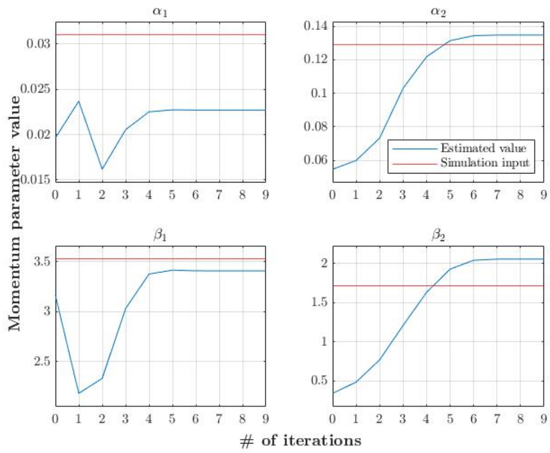

Figure 6: $\hat{\boldsymbol{\alpha}}_4$ and $\hat{\boldsymbol{\beta}}_4$ of selected attempt during projected NR method with $1e2$ initial guesses for $\hat{\boldsymbol{\theta}}_{4,0}$, a maximum number of iterations of $1e2$, a stopping criteria of $\boldsymbol{\epsilon} = 1e-2$ and $\delta = 5e-5$.

3.3.3 Discretized simulation

As demonstrated in figure 8 and table 2 (5.2.5.2) by writing out all (non-)Markovian components for $n = 1, 2, 3, 4$ discretized simulation does not properly estimate $\boldsymbol{P}$ sufficiently approximated via modified thinning simulation with $1e6$ realizations.

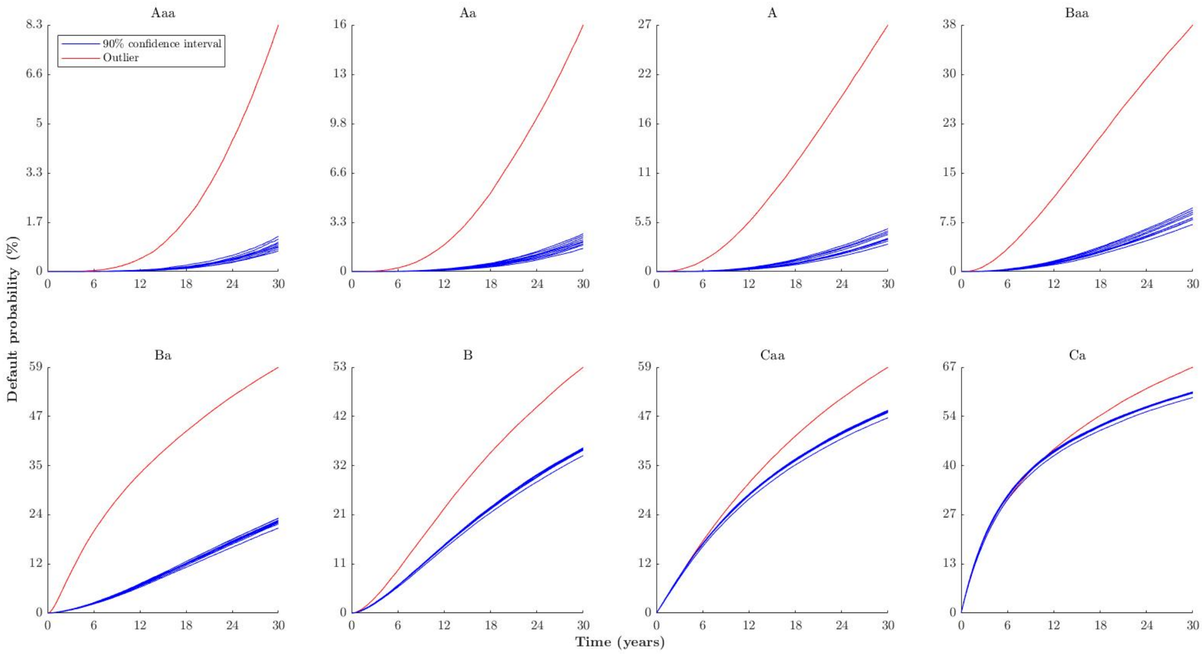

Figure 7: Bootstrapped default probabilities sufficiently approximated by modified thinning with $1e6$ realizations after application of projected NR method with 1 initial guess equal to $\hat{\boldsymbol{\theta}}_4$ based on the entire population for $\hat{\boldsymbol{\theta}}_0$, a maximum number of iterations of $1e2$, a stopping criteria of $\epsilon = 1e{-}2$ and $\delta = 5e{-}5$ on $1e1$ subsamples of size $1e4$ allowing for 90% confidence intervals.

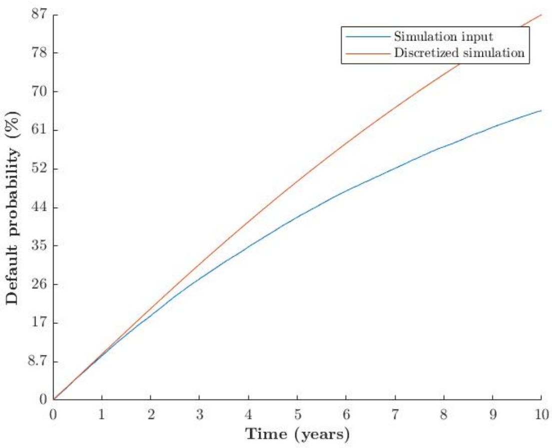

Figure 8: Default probabilities according to discretized simulation attempt and simulation input sufficiently approximated by modified thinning with $1e6$ realizations for initial state 1 with $\boldsymbol{S} = \{1, 2, 3\}$ and some $\boldsymbol{\theta}$.

3.3.4 Default probabilities

Overall PD for any initial state $i \in \boldsymbol{S}$ is compared in figure 9 with the true PD according to $\boldsymbol{P}$ sufficiently approximated via modified thinning simulation $1e6$ realizations. Similarly, modified thinning simulation allows the computation of $\hat{\boldsymbol{P}}_3$ and $\hat{\boldsymbol{P}}_4$ for the parameter estimations according to the MH algorithm and projected NR method.

3.3.5 Model expansions

An important aspect to consider for further research on the non-Markovian model is the possibility to be expanded and incorporate essential economic drivers, which are currently missing, e.g., correlation between ratings, business cycles, upgrade momentum or more granularity. An example of an expanded intensity function incorporating the above mentioned phenomena could be defined as follows

$$\begin{aligned}\lambda(t) = &\sum_{i=1}^{h-1} \mathbb{1}_{\{X(t)=i\}} q_i (1 + \gamma_i \sin(\delta t + \epsilon)) \\ &+ \sum_{n}^{N} \sum_{m=1}^{h-1} \sum_{\tau_{m,d} \in \boldsymbol{\tau}_{m,d}(t)} \rho_n \alpha_{m,d} e^{-\beta_{m,d}(t-\tau_{m,d})} \\ &+ \sum_{n}^{N} \sum_{m=1}^{h-1} \sum_{\tau_{m,u} \in \boldsymbol{\tau}_{m,u}(t)} \rho_n \alpha_{m,d} e^{-\beta_{m,d}(t-\tau_{m,u})}\end{aligned}$$

for some business cycle scale parameter per rating $\boldsymbol{\gamma} \in \mathbb{R}^h$, business cycle period parameter $\delta > 0$, business cycle phase shift parameter $\epsilon > 0$ and correlation parameter $\rho_n \in [-1, 1]$.

4 Discussion & conclusion

First, both the MH algorithm and projected NR method are reasonably accurate in terms of resulting $\hat{\boldsymbol{P}}_3$ and $\hat{\boldsymbol{P}}_4$ compared to $\hat{\boldsymbol{P}}$ as demonstrated in figure 9 (5.3.5), while the PDs for initial state Aaa

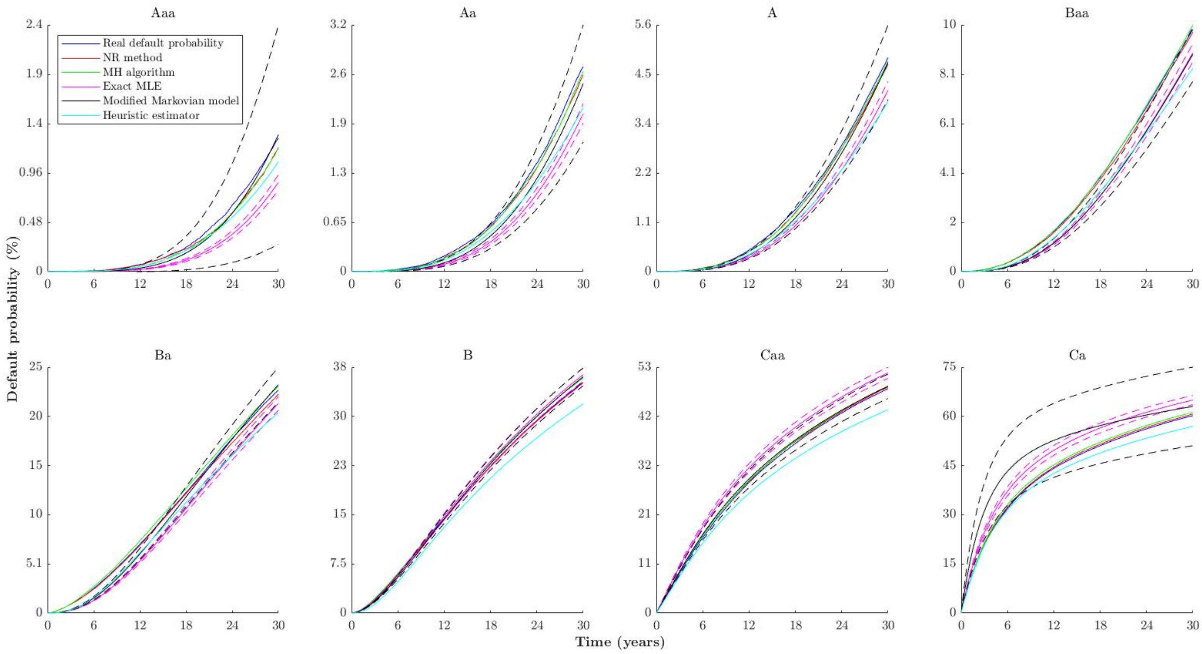

Figure 9: Default probabilities according to exact MLE, heuristic estimator, modified Markovian model and modified thinning simulation following parameter estimations via the MH algorithm and projected NR method sufficiently approximated by $1e6$ realizations. All parameter methodologies are applied as described in the theory (4) and results (5). As $\hat{\boldsymbol{P}}_1$ and $\hat{\boldsymbol{P}}_2$ are similar, the EM algorithm has been left out for clarity purposes.

show significant deviation over time horizon [0,T]. This approximation error of $\hat{\boldsymbol{P}}$ is probably caused by using a finite dataset with $\hat{\boldsymbol{\theta}}$ not in the neighbourhood of $\boldsymbol{\theta}$. Larger datasets, i.e., more initial ratings per state, theoretically drive better approximations of $\hat{\boldsymbol{P}}$ especially for the initial state Aaa as demonstrated in figure 11 (5.1). Although the results from both the MH algorithm and the projected NR method are already reasonably accurate, their precision can be improved by increasing either the number of iterations from $1e3$ to $1e4$, as is done in recent research, or initial guesses from $2e1$ to $1e1$ respectively [21]. The used setting with regards to ϕ, σ, $\hat{\boldsymbol{\theta}}_0$, $\boldsymbol{\epsilon}$ and $\boldsymbol{\delta}$ seems appropriate in connection to the results.

Second, $\hat{\boldsymbol{\theta}}_{3,0}$ is sampled close to $\hat{\boldsymbol{\theta}}$ as is shown in figures 4 (3.3.1), 12 and 13 (5.1) during the MH algorithm. The projected NR method is less dependent on a single value for $\hat{\boldsymbol{\theta}}_0$ and has converged with a significant deviation among values for $\hat{\boldsymbol{\theta}}_0$ and therefore might be considered more robust in practice.

Third, the MH algorithm and projected NR method differ significantly in terms of efficiency, whereas in this research the computational time is reduced from 6 to 1 hour by replacing the parameter estimation methodology. As the MH algorithm is demonstrated with $\hat{\boldsymbol{\theta}}_{3,0}$ close to $\hat{\boldsymbol{\theta}}_r$ it might be that the number of iterations needs to be increased from $1e3$ to $1e4$, as in recent research, to ensure convergence in general. The required computational time would then be approximately 60 hours, however, recent research was able to decrease the computational time needed for $1e4$ iterations with a $1e3$ burn-in to 8.5 hours [21]. Other than comparing the required computational time for both parameter estimation methodology, there is not yet a framework to properly compare the MH algorithm and projected NR method in terms of the rate of convergence. The model complexity does not allow a theoretical rate of convergence of the MH algorithm to be determined, while multiple roots and convexity might disturb the quadratic convergence of the projected NR method as is demonstrated in table 1 and figure 6 (3.3.2).

Fourth, the last aspect of performance is the possibility of bootstrapping, which is tightly linked to the computational time required. As the delta method does not seem to be applicable to the non-Markovian model and the computational time required for parameter estimation of multiple subsamples of the simulated dataset of $14,193$ (or population) is impractically large for both parameter estimation methodologies, bootstrapping does not seem possible (5.2.4.4). However, for large enough subsamples (near the size of the entire population equal to $1e4$) $\hat{\boldsymbol{\theta}}$ is a strong candidate for $\hat{\boldsymbol{\theta}}_{4,0}$ during the projected NR method applied to each subsample, which decreases the required iterations and computational time. The bootstrapping results are demonstrated in figure 7 (3.3.2), however, for a more precise approximation of the confidence intervals of $\hat{\boldsymbol{P}}_4$, the number of subsamples should be increases significantly.

Next, as demonstrated in figure 8 (3.3.4) and table 2 (5.2.5.2), discretized simulation is not possible for the non-Markovian model as the cross-products cannot be captured using an iterative algorithm. It seems that all possible paths have to be considered separately (as might be expected). A next step, however, is to investigate if certain paths with negligible low probability might be disregarded. Additionally, paths with negligibly low non-Markovian contributions might be properly approximated using a Markovian model. These two steps possibly allow an iterative scheme to sufficiently approximate PDs according to the non-Markovian model.

Also, the idea behind the heuristic estimator based on Chapman-Kolmogorov forward equations does not lead to proper approximations of $\boldsymbol{P}$ as demonstrated in figure 9 (5.3.5) , which might imply that the assumptions of (i) linearity, (ii) vanishing momentum after an upgrade or (iii) defaults only a result of solely consecutive downgrades are too strong. Chapman-Kolmogorov equations do not seem to form a strong basis on which the PDs according to the non-Markovian model can be approximated.

Furthermore, the modified Markovian model is robust in terms of resulting TPM $\boldsymbol{P}_8$ compared to the sufficient approximation of the real TPM $\boldsymbol{P}$ and allows the delta method to compute exact confidence intervals. The modified Markovian model outperforms the MH algortihm and projected NR method in terms of speed with a duration of ~ 2 minutes. Disadvantages of the modified Markovian model are larger confidence intervals as the rating data subsets $\boldsymbol{X}^i$ are smaller than $\boldsymbol{X}$ for any $i \in \boldsymbol{S}$. The modified Markovian model might also perform differently if other non-Markovian phenomena are incorporated other than momentum as it does not have the flexibility to adjust for other types of stochastic behaviour.

Moreover, the non-Markovian model shows great flexibility to adjust for different stochastic behaviour incorporating other types of non-Markovian or time-inhomogenuous phenomena often encountered in ratings, like seasonality, correlation, business cycles or upgrade momentum. This flexibility might also allow the non-Markovian model to be applied in different fields, like seismology, using similar types of models, however, further adjustment might make the non-Markovian model even more complex influencing the accuracy of current parameter estimation methodologies as, e.g., the loglikelihood $l(\boldsymbol{\theta}|\boldsymbol{X})$ might have many more stationary points located near each other preventing proper convergence [13].

Additionally, throughout the application of all parameter methodologies, MLE consistency has to be assumed due to the use of an absorption state describing a defaulting possibility for of both the (non-)Markovian models. Even so, the delta method cannot be applied to the non-Markovian model as it is currently too complex to find a path dependent closed form expression for $p_{ij}(t)$ for any $i, j \in \boldsymbol{S}$.

Besides, solely the mathematical point of view, there are some possible flaws in the non-Markovian model with regards to financial aspects, like excluding restart probabilities after default, seasonality, correlation, upgrade momentum and business cycles. Additionally, merits from other types of models like hidden Markov models, trying to capture the fact that ratings are not monitored continuously, are also neglected in the current non-Markovian model. Additionally, after an upgrade, there is still downgrade momentum, which keep the recursive patterns in tact, however, might incorrectly capture momentum. A downgrade not only increases the probability of further downgrades, but also decreases the stability of ratings, which is might not be desirable.

Finally, an additional suggestion for further research is to reduce the required computational time for both the MH algorithm and projected NR method. As a result, the projected NR method can be efficient enough to allow for more precise parametric bootstrapping within manageable time frames.

5 Appendix

5.1 Results

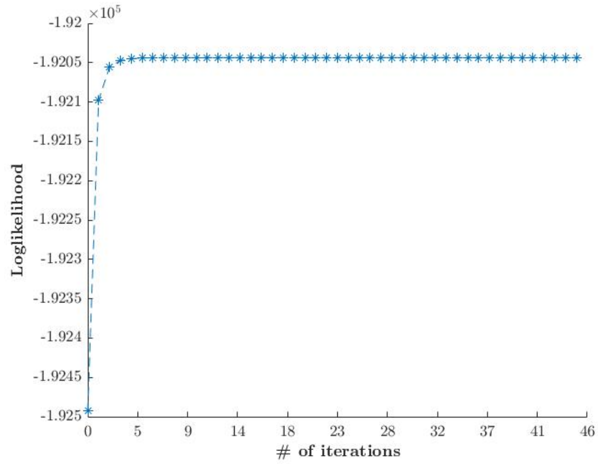

Figure 10: Loglikelihood during EM algorithm according to the Markovian model with stopping criteria $\epsilon = 1e-9$.

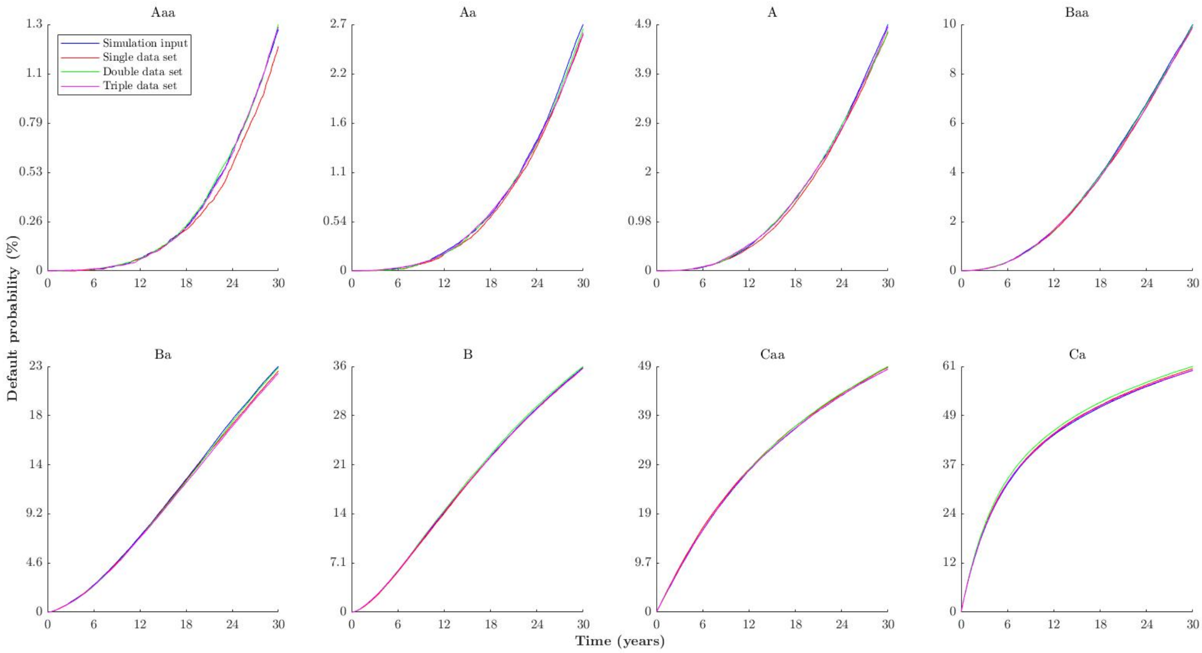

Figure 11: Default probabilities according to modified thinning simulation following parameter estimations via the projected NR method sufficiently approximated by modified thinning with $1e6$ realizations on rating datasets of 3 different sizes with $2e1$ initial guesses of $\boldsymbol{\theta}_{4,0}$, a maximum number of iterations of $1e2$, a stopping criteria of $\boldsymbol{\epsilon} = 1e-2$ and $\delta = 5e-5$.

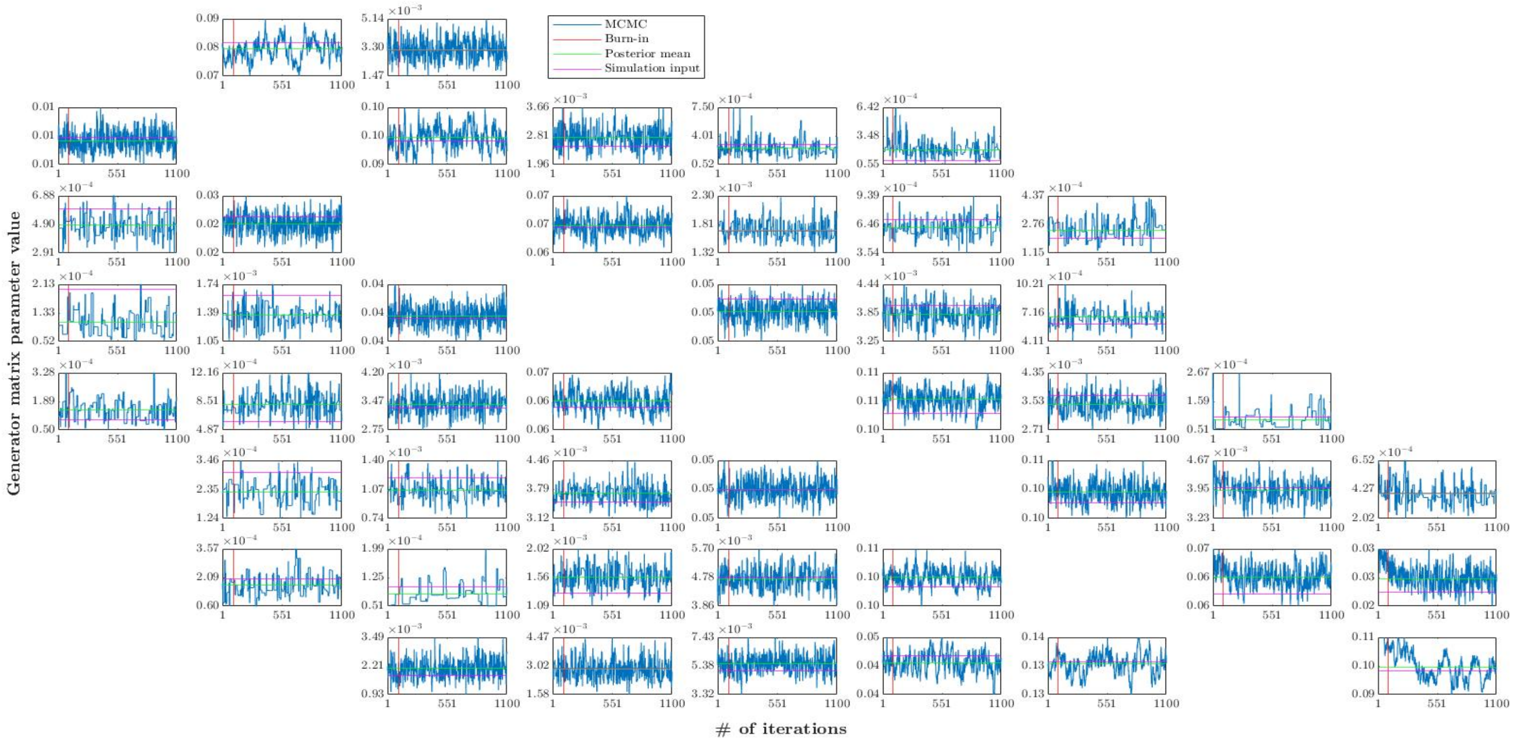

Figure 12: MCMCs of $\boldsymbol{V}_{\hat{\boldsymbol{Q}}_3}$ and corresponding posterior distributions, mean and real modified thinning simulation input for 1e3 iterations with 1e2 burn-in.

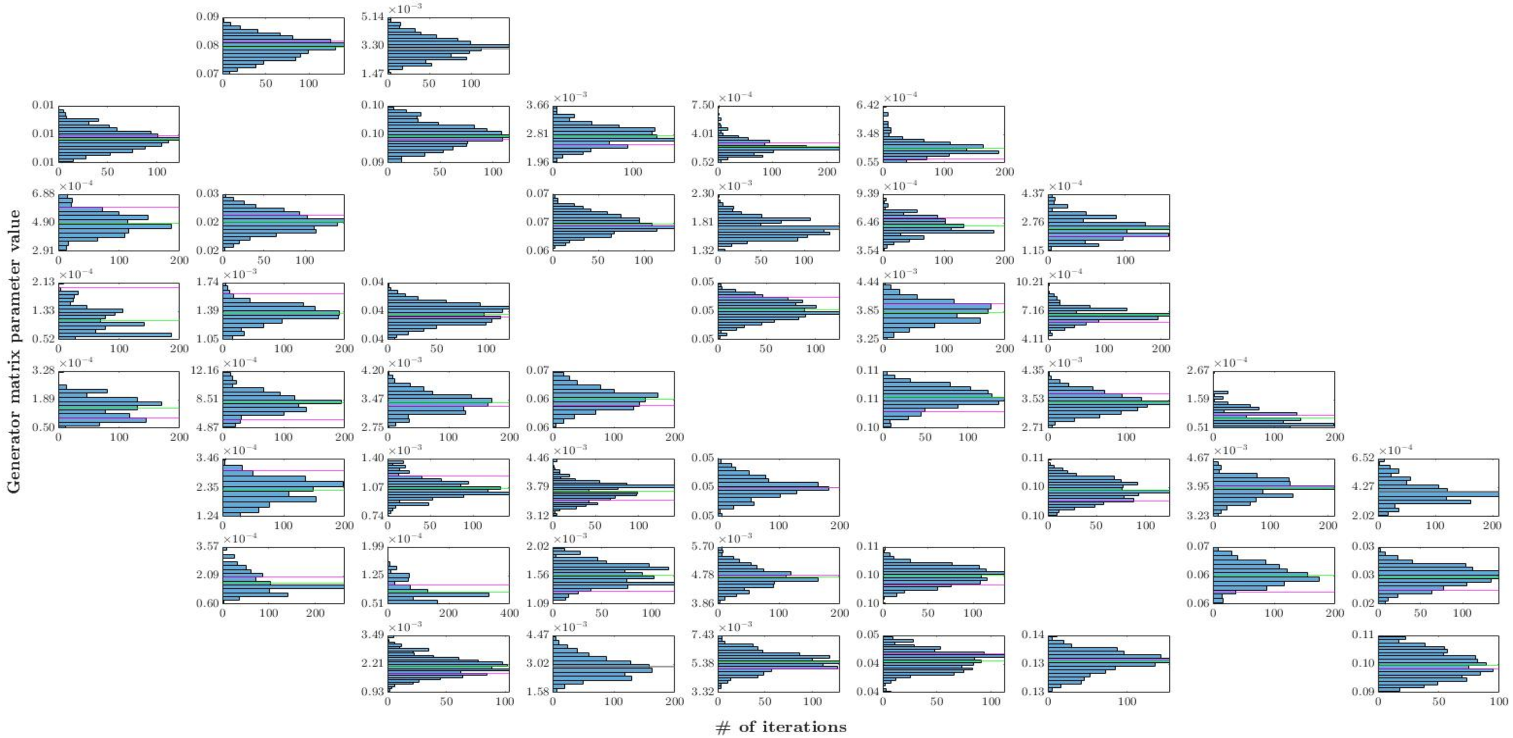

Figure 13: Histogram of $\boldsymbol{V}_{\hat{\boldsymbol{Q}}_3}$ and corresponding posterior distributions, mean and real modified thinning simulation input for 1e3 iterations with 1e2 burn-in.

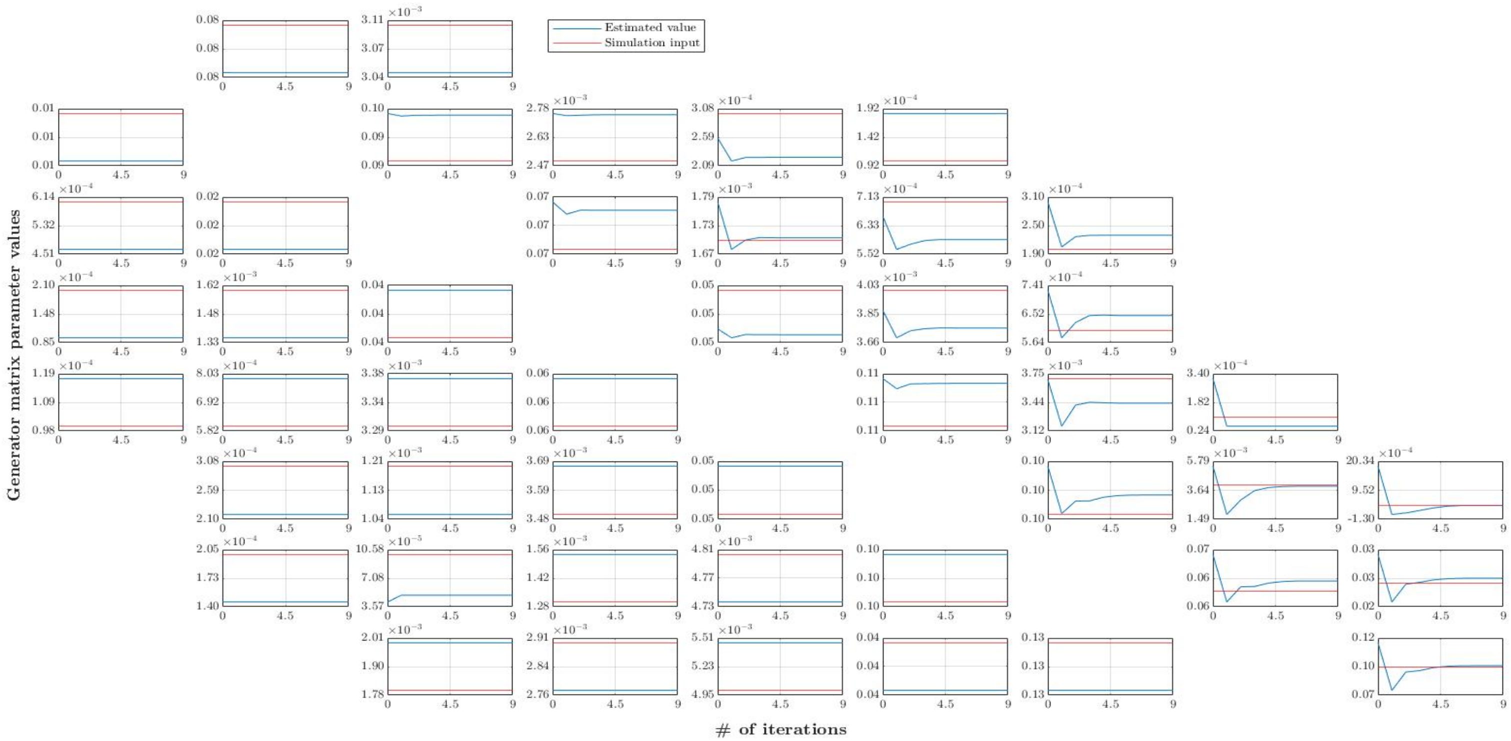

Figure 14: $\boldsymbol{V}_{\hat{\boldsymbol{Q}}_4}$ of selected attempt during projected NR method with $1e2$ initial guesses for $\hat{\boldsymbol{\theta}}_0$ with a maximum number of iterations of $1e2$, a stopping criteria of $\epsilon = 1e-2$ and $\delta = 5e-5$.

$$
\hat{\boldsymbol{Q}}^1 = \begin{pmatrix}
-8.59 & 8.38 & 0.21 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.30 & -10.60 & 9.15 & 0.12 & 0.03 & 0 & 0 & 0 & 0 \\
0.04 & 2.41 & -9.95 & 7.11 & 0.25 & 0.08 & 0.04 & 0 & 0 \\
0 & 0.08 & 3.47 & -8.49 & 4.61 & 0.16 & 0.16 & 0 & 0 \\
0 & 0 & 0.79 & 4.76 & -16.66 & 10.71 & 0.40 & 0 & 0 \\
0 & 0 & 0 & 0 & 4.09 & -16.36 & 9.54 & 2.04 & 0.68 \\
0 & 0 & 0 & 0 & 2.19 & 13.12 & -21.87 & 0 & 6.56 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -9.36 & 9.36 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \times 10^{-2}
$$

$$
\hat{\boldsymbol{Q}}_6^2 = \begin{pmatrix}
-8.03 & 7.82 & 0.21 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.05 & -10.99 & 9.64 & 0.29 & 0 & 0.01 & 0 & 0 & 0 \\
0.04 & 2.36 & -9.27 & 6.53 & 0.22 & 0.08 & 0.04 & 0 & 0 \\
0.01 & 0.12 & 4.08 & -9.82 & 5.01 & 0.50 & 0.10 & 0 & 0 \\
0.06 & 0.12 & 0.44 & 7.09 & -19.35 & 11.20 & 0.44 & 0 & 0 \\
0 & 0 & 0 & 0.25 & 3.80 & -14.85 & 9.90 & 0.74 & 0.17 \\
0 & 0 & 0 & 0.61 & 0 & 8.97 & -23.46 & 9.99 & 3.88 \\
0 & 0 & 0.67 & 0 & 0.67 & 4.71 & 14.80 & -30.28 & 9.42 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \times 10^{-2}
$$

$$
\hat{\boldsymbol{Q}}_6^3 = \begin{pmatrix}
-8.05 & 7.35 & 0.70 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.22 & -11.68 & 10.04 & 0.33 & 0.04 & 0.06 & 0 & 0 & 0 \\
0.05 & 2.32 & -9.33 & 6.74 & 0.16 & 0.04 & 0.03 & 0 & 0 \\
0.01 & 0.10 & 3.79 & -9.43 & 5.03 & 0.46 & 0.04 & 0 & 0 \\
0 & 0.06 & 0.36 & 6.32 & -16.96 & 9.84 & 0.34 & 0.04 & 0 \\
0 & 0 & 0.16 & 0.31 & 5.37 & -17.52 & 10.52 & 0.66 & 0.50 \\
0 & 0 & 0 & 0.08 & 0.41 & 11.27 & -23.45 & 7.46 & 4.23 \\
0 & 0 & 0 & 0.65 & 0.98 & 4.26 & 10.81 & -27.51 & 10.81 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \times 10^{-2}
$$

$$
\hat{\boldsymbol{Q}}_6^4 = \begin{pmatrix}
-11.72 & 10.36 & 1.35 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.30 & -10.85 & 9.09 & 0.38 & 0.08 & 0 & 0 & 0 & 0 \\
0.04 & 2.55 & -9.61 & 6.74 & 0.19 & 0.08 & 0.02 & 0 & 0 \\
0.01 & 0.15 & 3.82 & -9.24 & 4.83 & 0.35 & 0.070 & 0 & 0 \\
0 & 0.10 & 0.31 & 6.42 & -17.84 & 10.53 & 0.39 & 0.08 & 0 \\
0 & 0.04 & 0.07 & 0.30 & 5.31 & -16.72 & 9.98 & 0.72 & 0.30 \\
0 & 0 & 0 & 0.23 & 0.45 & 10.05 & -21.09 & 6.88 & 3.49 \\
0 & 0 & 0.23 & 0.12 & 0.58 & 3.72 & 11.04 & -24.86 & 9.18 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \times 10^{-2}
$$

$$
\hat{\boldsymbol{Q}}_6^5 = \begin{pmatrix}
-11.23 & 11.23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.98 & -10.79 & 8.37 & 0.22 & 0.22 & 0 & 0 & 0 & 0 \\
0.07 & 2.50 & -9.82 & 7.02 & 0.12 & 0.10 & 0 & 0 & 0 \\
0 & 0.13 & 3.95 & -9.18 & 4.61 & 0.39 & 0.10 & 0 & 0 \\
0.01 & 0.09 & 0.33 & 6.41 & -18.06 & 10.85 & 0.36 & 0.01 & 0 \\
0 & 0.04 & 0.07 & 0.39 & 4.84 & -17.22 & 10.87 & 0.67 & 0.35 \\
0 & 0.02 & 0.02 & 0.15 & 0.54 & 10.56 & -21.74 & 7.12 & 3.35 \\
0 & 0 & 0.13 & 0.50 & 0.50 & 3.89 & 14.23 & -31.78 & 12.54 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \times 10^{-2}
$$

$$
\hat{\boldsymbol{Q}}_6^6 = \begin{pmatrix}
-3.01 & 1.51 & 1.51 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.39 & -7.92 & 7.40 & 0.13 & 0 & 0 & 0 & 0 & 0 \\
0.03 & 2.06 & -8.92 & 6.38 & 0.18 & 0.18 & 0.10 & 0 & 0 \\
0.01 & 0.17 & 4.16 & -9.66 & 4.91 & 0.33 & 0.08 & 0 & 0 \\
0.01 & 0.06 & 0.37 & 6.51 & -18.52 & 11.19 & 0.36 & 0.01 & 0 \\
0 & 0.01 & 0.12 & 0.38 & 5.10 & -16.50 & 10.32 & 0.46 & 0.11 \\
0 & 0.01 & 0 & 0.14 & 0.40 & 10.57 & -21.67 & 7 & 3.53 \\
0 & 0 & 0.19 & 0.25 & 0.40 & 4.35 & 13.47 & -30.21 & 11.55 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \times 10^{-2}
$$

$$\hat{\boldsymbol{Q}}_6^7 = \begin{pmatrix} -19.28 & 14.46 & 4.82 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.76 & -11.47 & 9.70 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.16 & 1.64 & -7.87 & 5.90 & 0.16 & 0 & 0 & 0 & 0 \\ 0.06 & 0.12 & 3.59 & -9.25 & 5.17 & 0.30 & 0 & 0 & 0 \\ 0.04 & 0.04 & 0.18 & 5.95 & -17.85 & 11.16 & 0.41 & 0.07 & 0 \\ 0 & 0.04 & 0.09 & 0.34 & 4.91 & -16.43 & 10.25 & 0.64 & 0.15 \\ 0 & 0.03 & 0 & 0.15 & 0.58 & 10.23 & -20.52 & 6.68 & 2.87 \\ 0 & 0 & 0.28 & 0.16 & 0.80 & 4.82 & 13.69 & -31.87 & 12.12 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \times 10^{-2}$$

$$\hat{\boldsymbol{Q}}_6^8 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.73 & 1.73 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.99 & -9.98 & 4.99 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6.02 & -18.05 & 11.03 & 1 & 0 & 0 \\ 0 & 0 & 0.36 & 0.72 & 3.24 & -14.03 & 9.36 & 0 & 0.36 \\ 0 & 0 & 0 & 0.44 & 2.22 & 11.97 & -23.50 & 6.65 & 2.22 \\ 0 & 0 & 0 & 1.04 & 1.04 & 4.67 & 14.52 & -37.87 & 16.60 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \times 10^{-2}$$

5.2 Derivations

5.2.1 Markovian model

5.2.1.1 Markov property

$\boldsymbol{X}(t)$ satisfying the Markovian property for any $i, j \in \boldsymbol{S}$ and $t_1, t_2 \in [0, T]$ with $t_1 \leq t_2$ is defined by [30]

$$\mathbb{P}(X(t_2) = i | \mathcal{F}_{t_1}) = \mathbb{P}(X(t_2) = i | X_{t_1} = j)$$

5.2.1.2 Time-homogeneity

$\boldsymbol{X}(t)$ satisfying the time-homogeneity property for any $i, j \in \boldsymbol{S}$ and $t_1, t_2 \in [0, T]$ with $\Delta t > 0$ such that $t_1 + \Delta t, t_2 + \Delta t \in [0, T]$ is defined by [30]

$$\begin{aligned} \mathbb{P}(X(t_1 + \Delta t) = i | \mathcal{F}_{t_1}) &= \mathbb{P}(X(t_1 + \Delta t) = i | X(t_1) = j) && \text{Markovian property} \\ &= \mathbb{P}(X(t_2 + \Delta t) = i | X(t_2) = j) && \text{Time-homogeneity} \end{aligned}$$

5.2.1.3 Holding times

First, define the holding times $\{T_i > t\}$ and $\{T_i > t+\Delta t\}$ as $\{X(\bar{t}) = i : \bar{t} \in [0, t]\}$ and $\{X(\bar{t}) = i : \bar{t} \in [0, t+\Delta t]\}$ respectively for any $i \in \boldsymbol{S}$, $t \in [0, T]$ and $\Delta t > 0$ with $t + \Delta t \in [0, T]$. $\boldsymbol{X}(t)$ satisfying the Markovian and time-homogeneity properties implies T_i also holds these properties as follows [52]

$$\begin{aligned} \mathbb{P}(T_i > t + \Delta t | T_i > t) &= \mathbb{P}(X(\bar{t}) = i : \bar{t} \in [0, t + \Delta t] | X(\bar{t}) = i : \bar{t} \in [0, t]) && \text{Definition} \\ &= \mathbb{P}(X(\bar{t}) = i : \bar{t} \in [t, t + \Delta t] | X(\bar{t}) = i : \bar{t} \in [0, t]) && \text{Adapted} \\ &= \mathbb{P}(X(\bar{t}) = i : \bar{t} \in [t, t + \Delta t] | X(t) = i) && \text{Markovian property} \\ &= \mathbb{P}(X(\bar{t}) = i : \bar{t} \in [0, \Delta t] | X(0) = i) && \text{Time-homogeneity} \\ &= \mathbb{P}(T_i > \Delta t) && \text{Definition} \end{aligned}$$

Second, define the survival function, $S_i(t) = \mathbb{P}(T_i > t)$, for any $i \in \boldsymbol{S}$, and the $\mathbb{P}(T_i > t + \Delta t | T_i > t)$ according to Bayesian theorem by the following relation [34] [39]

$$\begin{aligned} \mathbb{P}(T_i > t + \Delta t | T_i > t) &= \frac{\mathbb{P}(T_i > t + \Delta t)}{\mathbb{P}(T_i > t)} && \text{Bayesian theorem} \\ &= \mathbb{P}(T_i > \Delta t) && \text{Markovian property \& time-homogeneity} \end{aligned}$$

implying for any $x, t \in [0, T]$ and $\Delta t > 0$ with $x = t + \Delta t$ by setting $\lambda_i = -\ln(S_i(1)) > 0$ that $S_i(x)$ is given by

$$S_i(x) = S_i(t + \Delta t) = S_i(t) S_i(\Delta t) = S_i(1)^{t+\Delta t} = S_i(1)^x = e^{\ln(S_i(1))x} = e^{-\lambda_i x}$$

such that T_i are IID exponential random variables with parameter $\lambda_i > 0$ as follows [34]

$$f_{T_i}(x) = -\frac{dS_i(x)}{dx} = -\frac{d(e^{-\lambda_i x})}{dx} = \lambda_i \cdot e^{-\lambda_i x}$$

5.2.1.4 Conditional transition probabilities

Define $N(t)$ as a counting process with holding times IID exponentially random variables driven by parameter $\lambda > 0$ over time horizon $[0, t]$ for any $t > 0$. Divide $[0, t]$ in $n \in \mathbb{N}^+$ equal bins with length $\Delta t = \frac{t}{n}$ such that for $x + y = \Delta t$ an upper bound for the probability of multiple transitions per bin is given by

$$\begin{aligned}
\mathbb{P}(k_b \geq 2) \leq \mathbb{P}(k_b = 2) &= \int_0^x \lambda e^{-\lambda z} dz \int_0^y \lambda e^{-\lambda z} dz && \text{Definition} \\
&= (1 - e^{-\lambda x})(1 - e^{-\lambda(\Delta t - x)}) && \text{-} \\
&\leq (1 - e^{-\lambda \frac{\Delta t}{2}})^2 && \text{Upper bound}
\end{aligned}$$

and according to a Maclaurin series expansion of $f(x) = e^x$ asymptotically equal to 0 as follows [1]

$$\lim_{\Delta t \to 0} \mathbb{P}(k_b \geq 2) \leq \lim_{\Delta t \to 0} (1 - e^{-\lambda \frac{\Delta t}{2}})^2 = \lim_{\Delta t \to 0} 1 - 2(1 - \frac{\lambda \Delta t}{2}) + (1 - \lambda \Delta t) + \mathcal{O}((\lambda \Delta t)^2) = 0$$

The probability of a single transition per bin is defined by

$$\mathbb{P}(k_b = 1) = \lim_{\Delta t \to 0} 1 - \mathbb{P}(k_b = 0) = \lim_{\Delta t \to 0} 1 - \int_{\Delta t}^{\infty} \lambda \cdot e^{-\lambda x} dx = \lim_{\Delta t \to 0} 1 - e^{-\lambda \Delta t}$$

which is according to a Maclaurin series expansion of $f(x) = 1 - e^{-x}$ asymptotically equal to $\lambda \Delta t$ as follows [1]

$$\lim_{\Delta t \to 0} f(\lambda \Delta t) = \lim_{\Delta t \to 0} f(0) + f'(0)\lambda \Delta t + \mathcal{O}((\lambda \Delta t)^2) = \lim_{\Delta t \to 0} \lambda \Delta t + \mathcal{O}((\lambda \Delta t)^2) = \lambda \Delta t$$

$N(t)$ asymptotically follows a binomial distribution with k transitions and $\binom{n}{k} = n!/(n-k)!$ the binomial coefficient by setting $p = \lambda \Delta t$ as follows [27]

$$\begin{aligned}
\lim_{n \to \infty} \mathbb{P}(N(t) = k) &= \lim_{n \to \infty} \binom{n}{k} \times \left(\lambda \frac{t}{n}\right)^k \times \left(1 - (\lambda \frac{t}{n})\right)^{n-k} && \text{Definition} \\
&= \lim_{n \to \infty} \left(\frac{(\lambda t)^k}{k!}\right) \times \left(\frac{n!}{(n-k)!} \frac{1}{n^k}\right) \times \left((1 - \frac{\lambda t}{n})^n\right) \times \left((1 - \frac{\lambda t}{n})^{-k}\right) && \text{Rearranged}
\end{aligned}$$

First, the following relation holds

$$\begin{aligned}
\lim_{n \to \infty} \frac{n!}{(n-k)!} \frac{1}{n^k} &= \lim_{n \to \infty} \frac{n(n-1)(n-2)\cdots 1}{(n-k)(n-k-1)(n-k-2)\cdots 1} \left(\frac{1}{n^k}\right) && \text{Definition} \\
&= \lim_{n \to \infty} \left(\frac{n}{n}\right) \times \left(\frac{n-1}{n}\right) \times \left(\frac{n-2}{n}\right) \times \cdots \times \left(\frac{n-k+1}{n}\right) && \text{Rearranged} \\
&= 1 && \text{Asymptotically}
\end{aligned}$$

Second, by setting $x = \frac{n}{\lambda t}$ according to the definition of $e = \lim_{x \to \infty} (1 + \frac{1}{x})^x$ the following relation holds [54]

$$\begin{aligned}
\lim_{n \to \infty} (1 - \frac{\lambda t}{n})^n &= \lim_{x \to \infty} \left((1 - \frac{1}{x})^x\right)^{\lambda t} && \text{Rearranged} \\
&= \lim_{x \to \infty} \left(\frac{1}{(1 + \frac{1}{x-1})^{x-1}(1 + \frac{1}{x-1})}\right)^{\lambda t} && \text{-} \\
&= \lim_{x \to \infty} \left(\frac{1}{(1 + \frac{1}{x})^x}\right)^{\lambda t} && \text{-} \\
&= e^{-\lambda t} && \text{Definition}
\end{aligned}$$

Third, for any $k > 0$ the following relation holds

$$\lim_{n \to \infty} (1 - \frac{\lambda t}{n})^{-k} = 1$$

such that for $n \to \infty$ $N(t)$ is asymptotically Poisson distributed as follows

$$\lim_{n \to \infty} \mathbb{P}(N(t) = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

For infinitesimal Δt the probability of a single transition by $\boldsymbol{X}(t)$ out of state $i \in \boldsymbol{S}$ with $q_i > 0$ is defined by [30]

$$\begin{aligned}
\mathbb{P}(X(t) \neq i | X(t - \Delta t) = i) &= \mathbb{P}(X(\Delta t) \neq i | X(0) = i) && \text{Time-homogeneity} \\
&= q_i \Delta t \cdot e^{-q_i \Delta t} && \text{Definition}
\end{aligned}$$

and according to a Maclaurin series expansion of $f(x) = e^x$ asymptotically equal to $q_i\Delta t + \mathcal{O}(\Delta t)$ as follows [1] [30] [35] [44]

$$\begin{aligned} q_i\Delta t \cdot e^{-q_i\Delta t} &= q_i\Delta t(1 - q_i\Delta t + \frac{q_i^2}{2!}\Delta t^2 - \frac{q_i^3}{3!}\Delta t^3 + \cdots && \text{Maclaurin series} \\ &= q_i\Delta t - (q_i\Delta t)^2 + \frac{q_i^3}{2!}\Delta t^3 - \frac{q_i^4}{3!}\Delta t^4 + \cdots && \text{Rearranged} \\ &= q_i\Delta t + \mathcal{O}(\Delta t) && \text{Asymptotically} \end{aligned}$$

5.2.1.5 Stable conservative generator matrix

The unconditional transition probability of $\boldsymbol{X}(t)$ for any $i, j \in \boldsymbol{S}$, infinitesimal Δt and $t \in [0, T]$ with $t + \Delta t \in [0, T]$ is defined by [30]

$$\begin{aligned} \mathbb{P}(X(t+\delta t) = j|X(t) = i) &= \mathbb{P}(X(\Delta t) = j|X(0) = i) && \text{Time-homogeneity} \\ &= \mathbb{P}(X(\Delta t) \neq i|X(0) = i) \\ &\times \mathbb{P}(X(\Delta t) = j|X(\Delta t) \neq i, X(0) = i) && \text{Bayesian theorem} \\ &= (q_i\Delta t + \mathcal{O}(\Delta t)) \times c && \text{Time-homogeneity} \end{aligned}$$

with $c \in [0, 1]$ as the time-homogeneity property implies that $\mathbb{P}(X(\Delta t) = j|X(\Delta t) \neq i, X(0) = i)$ is constant. Define the unnormalized conditional transition probability for any $i, j \in \boldsymbol{S}$ as follows [30] [35]

$$\begin{aligned} q_{ij} &= \lim_{\Delta t \to 0} \frac{\mathbb{P}(X(t) = j|X(t-\Delta t) = i)}{\Delta t} && \text{Definition} \\ &= \lim_{\Delta t \to 0} \frac{\mathbb{P}(X(\Delta t) = j|X(0) = i)}{\Delta t} && \text{Time-homogeneity} \end{aligned}$$

such that the conditional transition probability for any $i, j \in \boldsymbol{S}$ according to Bayesian theorem is asymptotically given by [30]

$$\begin{aligned} \lim_{\Delta t \to 0} \mathbb{P}(X(t) = j|X(t-\Delta t) = i, X(t) \neq i) &= \lim_{\Delta t \to 0} \frac{\mathbb{P}(X(t) = j|X(t-\Delta t) = i)}{\mathbb{P}(X(t) \neq i|X(t-\Delta t) = i)} && \text{Bayesian theorem} \\ &= \lim_{\Delta t \to 0} \frac{\mathbb{P}(X(\Delta t) = j|X(0) = i)}{\mathbb{P}(X(\Delta t) \neq i|X(0) = i)} && \text{Time-homogeneity} \\ &= \lim_{\Delta t \to 0} \frac{q_{ij}\Delta t + \mathcal{O}(\Delta t)}{q_i\Delta t + \mathcal{O}(\Delta t)} && \text{Definition} \\ &= \frac{q_{ij}}{q_i} && \text{-} \end{aligned}$$

and by the conservation of probability the following relation holds [30]

$$\sum_{j=1, j\neq i} \frac{q_{ij}}{q_i} = 1 \implies q_i = \sum_{j=1, j\neq i}^{h} q_{ij}$$

5.2.1.6 Transition probability matrix

According to a Maclaurins series expansion of $f(x) = e^x$ for any $i, j \in \boldsymbol{S}$ and infinitesmal Δt the TPM is defined by [1] [30]

$$\begin{aligned} (\boldsymbol{P}(t-\Delta t, t))_{ii} &= (\boldsymbol{P}(0, \Delta t))_{ii} && \text{Time-homogeneity} \\ &= 1 - q_i\Delta t + \mathcal{O}(\Delta t) && \text{Definition} \end{aligned}$$

and by the conservation of probability for $j \neq i$ the following relation holds [30]

$$(\boldsymbol{P}(0, \Delta t))_{ij} = q_i\Delta t\frac{q_{ij}}{q_i} + \mathcal{O}(\Delta t) = q_{ij}\Delta t + \mathcal{O}(\Delta t)$$

such that in matrix notation $\boldsymbol{P}$ is given by [30]

$$\boldsymbol{P}(t-\Delta t, t) = \boldsymbol{P}(0, \Delta t) = \mathbf{I}_h + \boldsymbol{Q}\Delta t + \mathcal{O}(\Delta t)$$

Set $t_1, t_2 \in [0, T]$ with $t_2 > t_1$, $n \in \mathbb{N}^+$ and $t_2 = t_1 + n\Delta t$, then $\boldsymbol{P}$ is defined by [30] [55]

$$\begin{aligned}
\boldsymbol{P}(t_1, t_2) &= \lim_{n\to\infty} (\mathbf{I} + \Delta t \cdot \boldsymbol{Q})^n && \text{Maclaurin} \\
&= \lim_{n\to\infty} \Big(\mathbf{I} + \frac{t_2 - t_1}{n}\boldsymbol{Q}\Big)^n && \text{Rearranged} \\
&= e^{\boldsymbol{Q}(t_2-t_1)} && \text{Definition} \\
&= \boldsymbol{P}(t_2 - t_1) && \text{Time-homogeneity}
\end{aligned}$$

5.2.2 Non-Markovian model

5.2.2.1 Markovian marked point process

The likelihood of a single MPP $\boldsymbol{X}$ over time horizon $[0, T]$ is defined by [17] [18] [21]

$$L(\lambda, f|\boldsymbol{X}) = \prod_{n=1}^{N(T)} \lambda(t_n) f(\kappa_n|t_n) e^{-\int_0^T \lambda(x)dx}$$

such that by setting the intensity as follows [21] [30]

$$\lambda(t_n) = \sum_{i=1}^{h} \mathbb{1}_{\{X(t_n)=i\}} q_i$$

with the following marks distribution [21] [30]

$$f(\kappa_n|t_n) = \sum_{i,j=1,j\neq i}^{h} \frac{\mathbb{1}_{\{X(t_n)=i,X(t_n^+)=j\}} q_{ij}}{q_i}$$

the Markovian model is given by

$$\begin{aligned}
L(\lambda, f|\boldsymbol{X}) = &\prod_{n=1}^{N(T)} \Big[\sum_{i=1}^{h} \mathbb{1}_{\{X(t_n)=i\}} q_i\Big] \times \Big[\sum_{i,j=1,j\neq i}^{h} \frac{\mathbb{1}_{\{X(t_n)=i,X(t_n^+)=j\}} q_{ij}}{q_i}\Big] \\
&\times \Big[e^{-\int_0^T \sum_{i=1}^{h} \mathbb{1}_{\{X(x)=i\}} q_i dx}\Big] && \text{Rearranging} \\
&= (q_i e^{-q_i t_1} \frac{q_{ij}}{q_i}) \times (q_j e^{-q_j(t_2-t_1)} \frac{q_{jk}}{q_j}) \times \cdots \times (q_k e^{-q_k(t_N-t_{N-1})} \frac{q_{kl}}{q_k}) \times e^{-q_l(T-t_N)} && \text{-}
\end{aligned}$$

5.2.2.2 Recursive patterns

The intensity is defined by an exponential Hawkes process as follows [21]

$$\lambda(t) = \sum_{i=1}^{h-1} \mathbb{1}_{\{X(t)=i\}} q_i + \sum_{\tau\in\boldsymbol{\tau}(t)} \alpha e^{-\beta(t-\tau)}$$

such that by setting a recursion as follows [21] [46]

$$\mathcal{R}_n = (\mathcal{R}_{n-1} + \mathbb{1}_{\{X(t_n^+)>X(t_n)\}}) e^{-\beta(t_n-t_{n-1})}$$

with $\mathcal{R}_0 = 0$ and $t_0 = 0$, $\lambda(t)$ is given by [21] [46]

$$\begin{aligned}
\lambda(t_{n+1}) &= \sum_{i=1}^{h-1} \mathbb{1}_{\{X(t_{n+1})=i\}} q_i + \alpha\mathcal{R}_{n+1} && \text{Rearranging} \\
&= \sum_{i=1}^{h-1} \mathbb{1}_{\{X(t_{n+1})=i\}} q_i + \sum_{\tau\in\boldsymbol{\tau}(t_{n+1})} \alpha e^{-\beta(t_{n+1}-\tau)} && \text{-} \\
&= \sum_{i=1}^{h-1} \mathbb{1}_{\{X(t_{n+1})=i\}} q_i + \alpha\Big(\sum_{\tau\in\boldsymbol{\tau}(t_n)} e^{-\beta(t_n-\tau)} + 1\Big) e^{-\beta(t_{n+1}-t_n)} && \text{-} \\
&= \sum_{i=1}^{h-1} \mathbb{1}_{\{X(t_{n+1})=i\}} q_i + \alpha\Big(\mathcal{R}_n + \mathbb{1}_{\{X(t_n^+)>X(t_n)\}}\Big) e^{-\beta(t_{n+1}-t_n)} && \text{-}
\end{aligned}$$

5.2.2.3 Loglikelihood

By filling the intensity [21]

$$\lambda(t) = \sum_{i=1}^{h-1} \mathbb{1}_{\{X(t)=i\}} q_i + \sum_{\tau \in \boldsymbol{\tau}(t)} \alpha e^{-\beta(t-\tau)}$$

and marks distribution [21]

$$f(\kappa_n|t_n) = \sum_{i,j=1, j\neq i}^{h} \frac{\mathbb{1}_{\{X(t_n)=i, X(t_n^+)=j\}}}{\lambda(t_n)} \times \Big(q_{ij} + \frac{\mathbb{1}_{X(t_n^+ > X(t_n))}}{\boldsymbol{G}_i} \sum_{\tau \in \boldsymbol{\tau}(t_n)} \alpha e^{-\beta(t_n-\tau)} \Big)$$

into the likelihood of a single MPP [17] [18] [21]

$$L(\lambda, f|\boldsymbol{X}) = \prod_{n=1}^{N(T)} \lambda(t_n) f(\kappa_n|t_n) e^{-\int_0^T \lambda(x)dx}$$

it follows that after cancelling out the $\lambda(t_n)$-terms and writing out the following relation [21]

$$\int_0^{\bar{T}} \sum_{\tau \in \boldsymbol{\tau}(\boldsymbol{x})} \alpha e^{-\beta(x-\tau)} dx = \sum_{\tau \in \boldsymbol{\tau}(\bar{T})} \frac{\alpha}{\beta}(1 - e^{-\beta(\bar{T}-\tau)})$$

the likelihood is defined by [21]

$$\begin{aligned}
L(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}, \hat{\alpha}, \hat{\beta}|\boldsymbol{X}) &= \prod_{n=1}^{N(\bar{T})} \Bigg[\Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X_{t_n}=i, X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{\{X(t_n^+ > X(t_n))\}}}{\boldsymbol{G}_i} \sum_{\tau \in \boldsymbol{\tau}(t_n)} \hat{\alpha} e^{-\hat{\beta}(t_n-\tau)} \Big) \Bigg] && \text{Definition} \\
&\times \exp\Big(- \int_0^T \sum_{i=1}^{h-1} \mathbb{1}_{X(x)=i} q_i + \sum_{\tau \in \boldsymbol{\tau}(x)} \hat{\alpha} e^{-\hat{\beta}(x-\tau)}) dx \Big) \\
&= \prod_{n=1}^{N(\bar{T})} \Bigg[\Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X_{t_n}=i, X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{\{X(t_n^+ > X(t_n))\}}}{\boldsymbol{G}_i} \sum_{\tau \in \boldsymbol{\tau}(t_n)} \hat{\alpha} e^{-\hat{\beta}(t_n-\tau)} \Big) \Bigg] && \text{Rearranging} \\
&\times \exp\Big(- \int_0^T \sum_{i=1}^{h-1} \mathbb{1}_{X(x)=i} q_i dx \Big) \times \exp\Big(- \sum_{\tau \in \boldsymbol{\tau}(\bar{T})} \frac{\hat{\alpha}}{\hat{\beta}} (1 - e^{-\hat{\beta}(\bar{T}-\tau)}) \Big)
\end{aligned}$$

such that the loglikelihood is given by [21]

$$\begin{aligned}
l(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}, \hat{\alpha}, \hat{\beta}|\boldsymbol{X}) &= \sum_{n=1}^{N(\bar{T})} \log \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X_{t_n}=i, X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+) > X(t_n))}}{\boldsymbol{G}_i} \sum_{\tau \in \boldsymbol{\tau}(t_n)} \hat{\alpha} e^{-\hat{\beta}(t_n-\tau)} \Big) \\
&- \int_0^{\bar{T}} \sum_{i=1}^{h-1} \mathbb{1}_{\{X(x)=i\}} q_i dx - \sum_{\tau \in \boldsymbol{\tau}(\bar{T})} \frac{\hat{\alpha}}{\hat{\beta}} (1 - e^{-\hat{\beta}(\bar{T}-\tau)})
\end{aligned}$$

The equality $q_i = -\sum_{j=1, j\neq i}^{h} q_{ij}$ implies the loglikelihood is defined by [21]

$$\begin{aligned}
l(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}, \hat{\alpha}, \hat{\beta}|\boldsymbol{X}) &= \sum_{n=1}^{N(\bar{T})} \log \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X_{t_n}=i, X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+) > X(t_n))}}{\boldsymbol{G}_i} \sum_{\tau \in \boldsymbol{\tau}(t_n)} \hat{\alpha} e^{-\hat{\beta}(t_n-\tau)} \Big) \\
&- \int_0^{\bar{T}} \sum_{i=1}^{h-1} \mathbb{1}_{\{X(x)=i\}} \Big(\sum_{j=1, j\neq i}^{h} q_{ij} \Big) dx - \sum_{\tau \in \boldsymbol{\tau}(\bar{T})} \frac{\hat{\alpha}}{\hat{\beta}} (1 - e^{-\hat{\beta}(\bar{T}-\tau)})
\end{aligned}$$

and by use of the following recursive patterns [21] [46]

$$\mathcal{R}_n = (\mathcal{R}_{n-1} + \mathbb{1}_{\{X(t_n^+) > X(t_n))\}}) e^{-\hat{\beta}(t_n - t_{n-1})}$$

with $\mathcal{R}_0 = 0$ and $t_0 = 0$ is given by [21] [46]

$$l(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}, \hat{\alpha}, \hat{\beta}|\boldsymbol{X}) = \sum_{n=1}^{N(\bar{T})} \log\Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X_{t_n}=i, X(t_n^+)=j\}} q_{ij} + \hat{\alpha} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{\boldsymbol{G}_i} \mathcal{R}_n \Big)$$
$$- \int_0^{\bar{T}} \sum_{i=1}^{h-1} \mathbb{1}_{\{X(x)=i\}} \Big(\sum_{j=1, j\neq i}^{h} q_{ij} \Big) dx - \sum_{\tau \in \boldsymbol{\tau}(\bar{T})} \frac{\hat{\alpha}}{\hat{\beta}} \Big(\sum_{n=1}^{N(\bar{T})} \mathbb{1}_{X(t_n^+)>X(t_n)} - \mathcal{R}_N e^{-\hat{\beta}(\bar{T}-\tau_N)} \Big)$$

for $\bar{T} = \min\{\{t : X(t) = h\} \wedge T\}$ and $N = N(T)$.

5.2.3 Parameter estimation Markovian model

5.2.3.1 Maximum likelihood estimator

For a single realization $\boldsymbol{X}$ over time horizon $[0, T]$ with transition times $\{t_1, t_2, ..., t_N\}$ for any $i, j, k, l \in \boldsymbol{S}$ and $N \in \mathbb{N}^+$ with $i \neq j$ and $j \neq k$ the likelihood is defined by [22] [30] [37]

$$\begin{aligned} L(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}|X) &= e^{-q_i t_1} q_{ij} e^{-q_j(t_2-t_1)} q_{jk} \cdots e^{-q_l(T-t_N)} && \text{Definition} \\ &= \prod_{i=1}^{h} \prod_{j\neq i}^{h} q_{ij}^{N_{ij}(T)} e^{-q_i R_i(T)} && \text{Rearranging} \end{aligned}$$

with $R_i(t) = \int_0^t \mathbb{1}_{\{X(\bar{t})=i\}} d\bar{t}$ the summed holding times in state i, $N_{ij}(t)$ the number of $i \to j$ transitions both up to time t and the last holding time following a censored exponential random variable [22] [30] [37]. According the equality $q_i = \sum_{i\neq j} q_{ij}$ the loglikelihood is defined by [22] [30] [37]

$$\begin{aligned} l(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}|X) &= \sum_{i=1}^{h} \sum_{i\neq j}^{h} \log(\hat{q}_{ij}) N_{ij}(T) - \sum_{i=1}^{h} \hat{q}_i R_i(T) && \text{Definition} \\ &= \sum_{i=1}^{h} \sum_{i\neq j}^{h} \log(\hat{q}_{ij}) N_{ij}(T) - \sum_{i=1}^{h} \sum_{i\neq j}^{h} \hat{q}_{ij} R_i(T) && \text{Rearranging} \end{aligned}$$

and the MLE defined by setting the partial derivative with respect to $\hat{q}_{ij}$ equal to 0 as follows [22] [30] [37]

$$\frac{\partial l(\hat{\boldsymbol{Q}}|X)}{\partial \hat{q}_{ij}} = \frac{N_{ij}(T)}{\hat{q}_{ij}} - R_i(T) = 0 \implies \hat{q}_{ij} = \frac{N_{ij}(T)}{R_i(T)}$$

with corresponding Hessian matrix $\boldsymbol{H}_l(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}|\boldsymbol{X})$ for any $k_1 = h(i_1 - 1) + j_1$ and $k_2 = h(i_2 - 1) + j_2$ given by [21] [22]

$$\begin{aligned} (\boldsymbol{H}_l(\boldsymbol{V}_{\hat{\boldsymbol{Q}}}|\boldsymbol{X}))_{k_1 k_2} &= \frac{\partial}{\partial \hat{q}_{i_2 j_2}} \frac{\partial l(\hat{\boldsymbol{Q}}|X)}{\partial \hat{q}_{i_1 j_1}} \\ &= \frac{\partial}{\partial \hat{q}_{i_2 j_2}} \frac{\boldsymbol{N}_{ij}(T)}{\hat{q}_{ij}} - R_i(T) \\ &= -\frac{N_{ij}(T)}{\hat{q}_{i_1 j_1}^2}\Big|_{(i_1,j_1)=(i_2,j_2)} \end{aligned}$$

otherwise if $(i_1, j_1) \neq (i_2, j_2)$, then $(\boldsymbol{H}_l \boldsymbol{V}_{\hat{\boldsymbol{Q}}}|\boldsymbol{X}))_{k_1 k_2}$ is equal to 0.

5.2.3.2 Uniqueness transition probability matrix

As eigenvalues of matrices with continuous random variables as entries are continuous random variables themselves, the probability of $\hat{\boldsymbol{Q}}$ not having h distinct eigenvalues is 0 almost surely [42]. Assume a stable conservative generator matrix $\boldsymbol{Q} \in \mathbb{R}^{h\times h}$ is diagonizable with $\boldsymbol{P}(t) = e^{\boldsymbol{Q}t}$. $\log(\boldsymbol{P}(t)) = \boldsymbol{Q}t$ if $\boldsymbol{P}(t)$ is invertible with a sufficient condition that $\det(\boldsymbol{P}) \neq 0$ [29]. As $\boldsymbol{Q}t = \boldsymbol{\Lambda}\boldsymbol{V}\boldsymbol{\Lambda}^{-1}$ with $\boldsymbol{V}$ a diagonal matrix with h distinct eigenvalues as entries and $\boldsymbol{\Lambda}$ the corresponding eigenvectors the matrix exponential is defined by [29]

$$e^{\boldsymbol{Q}t} = \boldsymbol{I}_h + \boldsymbol{Q}t + \frac{(\boldsymbol{Q}t)^2}{2!} + \frac{(\boldsymbol{Q}t)^3}{3!} + \cdots$$

and for any $n \in \mathbb{N}^+$

$$(\boldsymbol{Q}t)^n = (\boldsymbol{\Lambda V \Lambda}^{-1})^n = (\boldsymbol{\Lambda V \Lambda}^{-1})^n = \boldsymbol{\Lambda V \Lambda}^{-1}\boldsymbol{\Lambda V \Lambda}^{-1} \cdots \boldsymbol{\Lambda V \Lambda}^{-1} = \boldsymbol{\Lambda V}^n\boldsymbol{\Lambda}^{-1}$$

implying

$$\begin{aligned} e^{\boldsymbol{Q}t} &= \boldsymbol{\Lambda V}^0\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Lambda V}^1\boldsymbol{\Lambda}^{-1} + \frac{\boldsymbol{\Lambda V}^2\boldsymbol{\Lambda}^{-1}}{2!} + \frac{\boldsymbol{\Lambda V}^3\boldsymbol{\Lambda}^{-1}}{3!} + \cdots && \text{Definition} \\ &= \boldsymbol{\Lambda}(\boldsymbol{I} + \boldsymbol{V} + \frac{\boldsymbol{V}^2}{2!} + \frac{\boldsymbol{V}^3}{3!} + \cdots)\boldsymbol{\Lambda}^{-1} && \text{Rearranging} \\ &= \boldsymbol{\Lambda} e^{\boldsymbol{V}}\boldsymbol{\Lambda}^{-1} && \text{-} \\ &= \boldsymbol{P}(t) && \text{Definition} \end{aligned}$$

As the exponential function is one-to-one $\mathbb{R} \mapsto (0, \infty)$ the eigenvalues $e^{\boldsymbol{V}}$ of $\boldsymbol{P}$ are distinct and strictly positive [54]. By definition $\det(\boldsymbol{P}(t))$ is the product of the eigenvalues of $\boldsymbol{P}(t)$ implying $\det(\boldsymbol{P}(t)) > 0$ [29].

5.2.3.3 Exact maximum likelihood estimator discrete data

Given the initial state of an embedded discrete rating chain with $N \in \mathbb{N}^+$ transitions $\{X_1, X_2, ..., X_{N+1}\}$ the probability of this realization is as follows [30]

$$\begin{aligned} \mathbb{P}(\{X_1, X_2, ..., X_n\}) &= \prod_{n=2}^{N} p_{X_{n-1}X_n} \\ &= \prod_{i=1}^{h}\prod_{j=1}^{h} p_{ij}^{N_{ij}} \\ &= L(\boldsymbol{P}|\boldsymbol{X}) \end{aligned}$$

with $p_{ij} = (\boldsymbol{P})_{ij}$ fully describing discrete rating chain behaviour and N_{ij} the number of $i \to j$ transitions in the embedded rating chain $\{X_1, X_2, ..., X_{N+1}\}$ for any $i, j \in \boldsymbol{S}$. Using the conservation of probability as follows [30]

$$\sum_{j=1}^{h} p_{ij} = 1$$

for any $i \in \boldsymbol{S}$ the loglikelihood is as follows by setting [30]

$$p_{i1} = 1 - \sum_{j=2}^{h} p_{ij}$$

such that

$$l(\boldsymbol{P}|\boldsymbol{X}) = \sum_{i=1,j=2}^{h} N_{ij}\log p_{ij} + N_{i1}\log(1 - \sum_{j=2}^{h} p_{ij})$$

for any $i \in \boldsymbol{S}$ by setting the derivative of the loglikelihood fuction with respect to p_{ij} equal to 0 for any $i, j \in \boldsymbol{S}$ with $j \neq 1$ the MLE is found as follows [30]

$$\begin{aligned} \frac{l(\boldsymbol{P}|\boldsymbol{X})}{p_{ij}} &= \frac{d\sum_{i=1,j=2}^{h} N_{ij}\log p_{ij} + N_{i1}\log(1 - \sum_{j=2}^{h} p_{ij})}{dp_{ij}} \\ &= \frac{N_{ij}}{p_{ij}} - \frac{N_{i1}}{1 - \sum_{j=2}^{h} p_{ij}} \\ &= \frac{N_{ij}}{p_{ij}} - \frac{N_{i1}}{p_{i1}} \\ &= 0 \end{aligned}$$

implying

$$\frac{p_{ij}}{pi1} = \frac{N_{ij}}{N_{i1}}$$

for any $j \in \boldsymbol{S}$ such that $p_{ij} \propto N_{ij}$ which leads to the MLE after normalization [30]

$$\hat{p}_{ij} = \frac{N_{ij}}{\sum_{j=1}^{h} N_{ij}}$$

5.2.3.4 Consistency maximum likelihood estimator

By definition of the infinitesmal generator matrix $\boldsymbol{Q}$ driving $\boldsymbol{X}(t)$ over time horizon $[0,T]$ the following definition holds [2]

$$q_{ij} = \frac{\mathbb{E}[N_{ij}(T)]}{\mathbb{E}[R_i(T)]}$$

for any $i, j \in \boldsymbol{S}$ as [2]

$$\mathbb{E}[R_i(T)] = \int_0^T \mathbb{P}(\boldsymbol{X}(t) = i)dt$$

and [2]

$$\mathbb{E}[N_{ij}(T)] = q_{ij} \int_0^T \mathbb{P}(\boldsymbol{X}(t) = i)dt$$

since for $\Delta t = \frac{T}{n}$ for some sufficiently large $n \in \mathbb{N}^+$ [2]

$$\begin{aligned}
\mathbb{E}[N_{ij}(T)] &= \sum_{k=0}^{n} \mathbb{1}_{\{\boldsymbol{X}((k+1)\Delta t)=j, \boldsymbol{X}(k\Delta t)=i\}} \\
&= \sum_{k=0}^{n} \mathbb{P}(\boldsymbol{X}((k+1)\Delta t) = j \wedge \boldsymbol{X}(k\Delta t) = i) + \mathcal{O}(\Delta t) \\
&= \sum_{k=0}^{n} q(i,j)\mathbb{P}(\boldsymbol{X}(k\Delta t) = i) + \mathcal{O}(\Delta t) \\
&\to q_{ij} \int_0^T \mathbb{P}(\boldsymbol{X}(t) = i)dt
\end{aligned}$$

as $n \to \infty$, since again per bin with length Δt only a single transition is considered with an error term of $\mathcal{O}(\Delta t)$. Define the MLE of $\boldsymbol{Q}$ after $m \in \mathbb{N}^+$ IID realizations as $\hat{q}_{ij}^m$ such that [2]

$$\lim_{m\to\infty} \hat{q}_{ij}^m = \lim_{m\to\infty} \frac{N_{ij}^m(T)/m}{R_i^m(T)/m} = q_{ij}$$

with in previous notation $N_{ij}^m(T) = N_{ij}(T)$ and $R_i^m(T) = R_i(T)$. Now, the set of random variables

$$\{\sqrt{m}(\hat{q}_{ij}^m - q_{ij})\} \sim \Big\{\frac{1}{\mathbb{E}[R_i(T)]}\Big(\frac{N_{ij}^m(T) - q_{ij}R_i^m(T)}{\sqrt{m}}\Big)\Big\}_{i\neq j}$$

as it is allowed to set $\mathbb{E}[R_i(T)] = \frac{R_i^m(T)}{k}$ as $m \to \infty$ by the extended version of the CLT [2] being asymptotically normal with mean 0 and covariances

$$\begin{aligned}
\Sigma_{i,j,k,l} &= \frac{1}{\mathbb{E}[R_i(T)]\mathbb{E}[R_k(T)]}\mathbb{E}[(N_{ij}^m(T) - q_{ij}R_i^m(T))(N_{kl}^m(T) - q_{kl}R_k^m(T))] \\
&= \delta((i,j),(k,l))\mathbb{E}[R_i(T)]
\end{aligned}$$

5.2.3.5 The delta method

The use of the Delta method does not generally hold as rating chains are not irreducible. It must be assumed that the MLE is a consistent estimator such that [38]

$$\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \xrightarrow{D} \mathcal{N}(0, \Sigma)$$

implying [38]

$$\sqrt{n}(\boldsymbol{P}(\hat{\boldsymbol{\theta}}) - \boldsymbol{P}(\boldsymbol{\theta})) \overset{D}{\Longrightarrow} \mathcal{N}(0, \mathrm{Var}(\boldsymbol{P}(\boldsymbol{\theta})))$$

First, apply a first order Taylor series on $\boldsymbol{P}$ around $\hat{\boldsymbol{\theta}}$ as follows [38]

$$\boldsymbol{P}(\hat{\boldsymbol{\theta}}) \approx \boldsymbol{P}(\boldsymbol{\theta}) + \nabla\boldsymbol{P}(\boldsymbol{\theta})^T(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})$$

with Σ the covariance matrix of $\hat{\boldsymbol{\theta}}$ implying that the variance is defined by [38]

$$\begin{aligned}\text{Var}(\boldsymbol{P}(\hat{\boldsymbol{\theta}})) &\approx \text{Var}(\boldsymbol{P}(\boldsymbol{\theta}) + \nabla\boldsymbol{P}(\boldsymbol{\theta})^T(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})) \\ &= \text{Var}(\boldsymbol{P}(\boldsymbol{\theta}) + \nabla\boldsymbol{P}(\boldsymbol{\theta})^T\hat{\boldsymbol{\theta}} - \nabla\boldsymbol{P}(\boldsymbol{\theta})^T\boldsymbol{\theta}) \\ &= \text{Var}(\nabla\boldsymbol{P}(\boldsymbol{\theta})^T\hat{\boldsymbol{\theta}}) \\ &= \nabla\boldsymbol{P}(\boldsymbol{\theta})^T\Sigma\nabla\boldsymbol{P}(\boldsymbol{\theta})\end{aligned}$$

as $\text{Var}(a + X) = \text{Var}(X)$ for any random variable X and constant $a \in \mathbb{R}$. Finally, the delta method implies as $\hat{\boldsymbol{\theta}}$ is assumed to be consistent that [38]

$$\sqrt{n}(\boldsymbol{P}(\hat{\boldsymbol{\theta}}) - \boldsymbol{P}(\boldsymbol{\theta})) \overset{D}{\Longrightarrow} \mathcal{N}(0, \nabla\boldsymbol{P}(\boldsymbol{\theta})^T\Sigma\nabla\boldsymbol{P}(\boldsymbol{\theta}))$$

5.2.4 Parameter estimation non-Markovian model

5.2.4.1 Convergence Metropolis-Hastings algorithm

Part I

The following proof on the convergence of the Metropolis-Hastings algorithm in general is simplified to a single-dimensional finite state parameter space $\boldsymbol{\Theta}$, which can be expanded to a multidimensional continuous setting [12].

Let $\{\theta_n\}_{n\geq 0}$ be a Markov chain sampled according the Metropolis-Hastings algorithm with some time-homogenuous proposal function $\psi(\theta_{n+1}|\theta_n)$ translating into a transition matrix $\boldsymbol{\Psi}$ satisfying the Markov property and $M \in \mathbb{N}^+$ dimensional finite state parameter space $\{\theta^1, ..., \theta^M\}$. Assuming $\{\theta_n\}_{n\geq 0}$ is (i) irreducible, (ii) aperiodic and (iii) there exists a stationary distribution, $\boldsymbol{\pi}$, such that $\boldsymbol{\pi} = \boldsymbol{\pi}\boldsymbol{\Psi}$, then according to the basic limit theorem [12]

$$\lim_{n\to\infty} \mathbb{P}(\theta_n = \theta^i) \lim_{n\to\infty} \pi_n(i) = \pi(i)$$

for any $i = 1, ..., N$ and any initial distribution π_0 of θ_0 also defined by the total variation as follows [12]

$$||\pi_n - \pi|| = \sup_{\mathcal{A}\in\Theta}(\pi_n(\mathcal{A}) - \pi(\mathcal{A}))$$

Part II

Define $\{\vartheta_n\}_{n\geq 0}$ as a Markov chain independent from $\{\theta_n\}_{n\geq 0}$ with similar $\boldsymbol{\Psi}$, but let ϑ_0 have initial distribution π as opposed to θ_0 having initial distribution π_0. Set the coupling time T as the first time θ_n equals ϑ_n [12]

$$T = \inf\{n : \theta_n = \vartheta_n\}$$

and define a new Markov chain as follows [12]

$$\bar{\vartheta} = \begin{cases} \vartheta_n & \text{if } n < T \\ \theta_n & \text{if } n \geq T \end{cases}$$

$\{\bar{\vartheta}_n\}_{n\geq 0}$ is stationary as $\bar{\vartheta}_0 = \vartheta_0$ with $\vartheta_0 \sim \pi$ and $\boldsymbol{\pi} = \boldsymbol{\pi}\Psi^n$ for any $n \in \mathbb{N}^+$ implying the the total variation is given by [12]

$$\begin{aligned}\pi_n(\mathcal{A}) - \pi(\mathcal{A}) &= \mathbb{P}(\theta_n \in \mathcal{A}) - \mathbb{P}(\bar{\vartheta}_n \in \mathcal{A}) \\ &= \mathbb{P}(\theta_n \in \mathcal{A}, n \geq T) - \mathbb{P}(\bar{\vartheta}_n \in \mathcal{A}, n \geq T) + \mathbb{P}(\theta_n \in \mathcal{A}, n < T) - \mathbb{P}(\bar{\vartheta}_n \in \mathcal{A}, n < T) \\ &= \mathbb{P}(\theta_n \in \mathcal{A}, n < T) - \mathbb{P}(\bar{\vartheta}_n \in \mathcal{A}, n < T) \\ &< \mathbb{P}(T > n)\end{aligned}$$

Proving convergence of the total variation $||\pi_n - \pi|| \to 0$ is equivalent to $\mathbb{P}(T > n) \to 0$ or $\mathbb{P}(T < \infty) = 1$ as $n \to \infty$ [12].

Part III

Define the bivariate Markov chain $\{\xi_n = (\theta_n, \vartheta_n)\}_{n\geq 0}$ such that the coupling time T is defined by the event that ξ_n hits the diagonal line $\{(\theta^i, \vartheta^i) : \theta^i = \vartheta^i \in \Theta\}$ with [12]

$$\mathbb{P}^{\xi}(\theta_{n+1} = \theta^j, \vartheta_{n+1} = \vartheta^j | \theta_n = \theta^i, \vartheta_n = \vartheta^i) = \mathbb{P}(\theta_{n+1} = \theta^j | \theta_n = \theta^i)\mathbb{P}(\vartheta_{n+1} = \vartheta^j | \vartheta_n = \vartheta^i)$$

for any $\theta^i, \theta^j \vartheta^i \vartheta^j \in \Theta$ as $\theta_n \perp \vartheta_n$ for any $n \in \mathbb{N}^+$ with stationary distributions

$$\pi^{\xi}(\theta^i, \vartheta^i) = \pi(\theta^i)\pi(\vartheta^i)$$

The proof $\mathbb{P}(T < \infty) = 1$ is now reduced to proving that ξ_n hits the diagonal $\{\xi_n = (\theta_n, \vartheta_n)\}_{n\geq 0}$ with probability 1. It suffices to proof that $\{\xi_n\}$ is irreducible and recurrent, however, a stationary distribution $\boldsymbol{\pi}$ satisfies the following condition [12]

$$\sum_i \boldsymbol{pi}(\theta_i)\boldsymbol{\Psi}^n(\theta^i, \theta^j) = \boldsymbol{\pi}(\theta^j)$$

for all n, such that if a state θ^j is not recurrent (transient) it holds that $\pi(\theta^j) = 0$. As irreducibility holds that $\boldsymbol{\Psi}(\theta^i, \theta^j) > 0$ for any $(\theta^i, \theta^j) \in \Theta$ by definition it implies for a stationary distribution with $\pi(\theta^j) = 0$ that $\pi(\theta^i) = 0$ for any $(\theta^i, \theta^j) \in \boldsymbol{\Theta}$ contradicting that $\boldsymbol{\pi}$ is a stationary distribution summing to 1 and showing that it suffices to proof that ξ_n is irreducible [12].

Part IV

To proof ξ_n is irreducible it must be proven that $\boldsymbol{\Psi}^n(\theta^i, \theta^j) > 0$ holds for sufficiently large n. This is done by proving that $\exists N \in \mathbb{N}^+$ such that $\forall n \geq N$ it holds that $n \in \{\bar{n} : \boldsymbol{\Psi}^{\bar{n}}(\theta^i, \theta^j)\}$. As $\{\bar{n} : \boldsymbol{\Psi}^{\bar{n}}(\theta^i, \theta^j)\}$ is aperiodic by assumption implying a greatest common divisor of 1 and is closed under addition it sufficies to proof that the above condition holds for any set $\mathcal{A}$, which is closed under addition and has a greatest common divisor 1 [12].

Let $\mathcal{A}$ be a set of integers closed under addition and with greatest common divisor equal to 1. First, it is proven that $\mathcal{A}$ contains at least one pair of consecutive integers via a contradiction. Assume the minimal distance $s > 1$ between 2 sequential elements of $\mathcal{A}$, then $\exists n_1$ such that $n_1, n_1 + s \in \mathcal{A}$. Set $n_2 \in \mathcal{A}$ as an integer not divided by s, which must exist, since the greatest common divisor is 1. Let $n_2 = ms + r$ with $0 < r < s$ and $m \in \mathbb{N}^+$ and define $n_3 = (m+1)(n_1 + s) \in \mathcal{A}$ and $n_4 = (m+1)n_1 + n_2 \in \mathcal{A}$ since $\mathcal{A}$ is closed under addition. Finally, $n_3 - n_4 = s - r \in (0, s)$ contradicting that the minimal spacing is $s > 1$ [12].

It is proven that ξ_n is irreducible for sufficiently large n such that $\mathbb{P}(T < \infty) = 1$ and it is proven that any irreducible and aperiodic Markov chain θ_n with a stationary distribution π converges towards this stationary distribution [12].

5.2.4.2 Convergence projected Newton-Raphson method

Part I

The setting with sufficient conditions to proof convergence to a root is demonstrated in one dimension, then generalized to a multidimensional case. Let f be a continuous twice differentiable loglikelihood function with $f : \mathbb{R} \mapsto [0, 1]$ and root $x \in \mathbb{R}$. Apply a second order Taylor series around an initial guess of the root $x_0 \in \mathbb{R}$ [31]

$$f(x_0 + \Delta x) \approx f(x_0) + f'(x_0)\Delta x + \frac{1}{2}f''(x_0)\Delta x^2$$

with $\Delta x \in \mathbb{R}$ and ξ_0 between x_0 and the root x. By taking the derivative with respect to Δx and setting equal to 0, $f(x_0 + \Delta x)$ is locally minimized or maximized over Δx as follows [5] [58]

$$\begin{aligned} 0 = \frac{d}{d\Delta x}(f(x_0 + \Delta x)) &\approx \frac{d}{d\Delta x}(\Delta x f(x_0) f'(x_0)) + \frac{\Delta x^2}{2} f''(x_0) \\ &= f'(x_0) + f''(x_0)\Delta x \end{aligned}$$

implying [5] [58]

$$\Delta x = -\frac{f'(x_n)}{f''(x_n)}$$

Finally, $f(x_0)$ is locally minimized or maximized using a second order Taylor series approximation by setting [5] [58]

$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)}$$

leading to a general iterative method by setting [5] [58]

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

Part II

Define the error $\epsilon_n = x - x_n$ with the root x and set a second order Taylor series of $f'(\cdot)$ around the root x equal to 0 as follows [5] [58]

$$f'(x) = f'(x_n) + f''(x_n)(x - x_n) + \frac{(x - x_n)^2}{2} f'''(\xi_n) = 0$$

with ξ_n between x_n and the root x. By dividing both sides by $f''(x_n)$ it holds that

$$\frac{f'(x_n)}{f''(x_n)} + (x - x_n) - \frac{f'''(x_n)}{2f''(x_n)}(x - x_n)^2 = 0$$

and by substitution according to the NR method as follows

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

such that the convergence of the NR method is at least quadratic according to [5] [58]

$$x - x_{n+1} - \frac{f'''(x_n)}{2f''(x_n)}(x - x_n)^2 = 0$$

or

$$\epsilon_{n+1} = -\frac{\epsilon_n^2 f'''(\xi_n)}{2f''(x_n)}$$

Error computation is made more robust by considering absolute values such that [5] [58]

$$|\epsilon_{n+1}| = \frac{\epsilon_n^2 |f'''(\xi_n)|}{2|f''(x_n)|}$$

Furthermore, the error can be defined as follows

$$|\epsilon_{n+1}| \leq M\epsilon_n^2$$

with

$$M = \sup_{\bar{x} \in (0,\infty)} \left| \frac{f'(\bar{x})}{f''(\bar{x})} \right|$$

such that an absolute upper bound for the error ϵ_n can be computed given an upper bound for ϵ_0. Based on this derivation, the convergence conditions for the NR method requires that $M|\epsilon_0| < 1$.

Part III

As the principle of a second order Taylor series remains given infinitesimal $\Delta\boldsymbol{\theta}$ for any $\boldsymbol{\theta}, \boldsymbol{\theta} + \Delta\boldsymbol{\theta} \in \boldsymbol{\Theta}$, the NR method can be extended to a multi dimensional setting by setting the gradient of $l(\boldsymbol{\theta}|\boldsymbol{X})$ equal to 0 as follows [5] [58]

$$\nabla l(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}|\boldsymbol{X}) \approx \nabla l(\boldsymbol{\theta}|\boldsymbol{X}) + \nabla^2 l(\boldsymbol{\theta})\Delta\boldsymbol{\theta} = 0$$

implying

$$\Delta\boldsymbol{\theta} = -(\nabla^2 l(\boldsymbol{\theta}|X))^{-1} \nabla l(\boldsymbol{\theta}|X)$$

such that

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - (\nabla^2 l(\boldsymbol{\theta}|X))^{-1} \nabla l(\boldsymbol{\theta}|X)$$

given that $l(\boldsymbol{\theta}|X)$ is twice differentiable and continuous for $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. Accordingly, the error, $\boldsymbol{\epsilon}$, is defined by [5] [58]

$$\boldsymbol{\epsilon}_{n+1} = -\frac{\boldsymbol{\epsilon}_n^2}{2}(\nabla^2 l(\boldsymbol{\theta}_n|\boldsymbol{X}))^{-1}\mathbf{D}^3 l(\boldsymbol{\xi}_n|\boldsymbol{X})$$

with $\boldsymbol{\xi}_n$ between $\boldsymbol{\theta}_n$ and the root $\boldsymbol{\theta}_r$ in any dimension and $\mathbf{D}^3$ an hyper matrix [5] [58] such that

$$\mathbf{D}^3 l(\boldsymbol{\xi}_n|\boldsymbol{X}) = \sum \frac{\partial^3 l(\boldsymbol{\xi}_n|\boldsymbol{X})}{\partial\boldsymbol{\theta}_i \partial\boldsymbol{\theta}_j \partial\boldsymbol{\theta}_k}\boldsymbol{\theta}_i\boldsymbol{\theta}_j\boldsymbol{\theta}_k$$

implying quadratic convergence. Due to complexity $\mathbf{D}^3 l(\boldsymbol{\xi}_n|\boldsymbol{X})$ is not attained and absolute errors can not be computed.

5.2.4.3 Partial derivatives loglikelihood

$$\begin{aligned} l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X}) = &\sum_{n=1}^{N(\bar{T})} \log\Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X_{t_n}=i, X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}(t_n)} \alpha e^{-\beta(t_n-\tau)}\Big) \\ &- \int_0^{\bar{T}} \sum_{i=1}^{h-1} \mathbb{1}_{\{X(x)=i\}}\Big(\sum_{j=1, j\neq i}^{h} q_{ij}\Big)dx - \sum_{\tau\in\boldsymbol{\tau}(\bar{T})} \frac{\alpha}{\beta}(1-e^{-\beta(\bar{T}-\tau)}) \end{aligned}$$

with

$$\mathcal{R}_n = (\mathcal{R}_{n-1} + \mathbb{1}_{\{X(t_n^+)>X(t_n))\}})e^{-\beta(t_n-t_{n-1})}$$

for $R_0 = 0$ and $t_0 = 0$ and $X(0^-) = X(0)$ implying the following partial derivatives

$$\begin{aligned} l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X}) = &\sum_{n=1}^{N(\bar{T})} \log\Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X_{t_n}=i, X(t_n^+)=j\}} q_{ij} + \alpha\frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i}\mathcal{R}_n\Big) \\ &- \int_0^{\bar{T}} \sum_{i=1}^{h-1} \mathbb{1}_{\{X(x)=i\}}\Big(\sum_{j=1, j\neq i}^{h} q_{ij}\Big)dx - \sum_{\tau\in\boldsymbol{\tau}(\bar{T})} \frac{\alpha}{\beta}\Big(\sum_{n=1}^{N(\bar{T})} \mathbb{1}_{X(t_n^+)>X(t_n)} - \mathcal{R}_N e^{-\beta(\bar{T}-\tau_N)}\Big) \end{aligned}$$

$$\begin{aligned} \frac{\partial l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial \alpha_m} = &\sum_{n=1}^{N(\bar{T})} \mathbb{1}_{\{\boldsymbol{\tau}_m(t_n)\neq\emptyset\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} e^{-\beta_m(t_n-\tau)} \\ &\times\Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n)=i, X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2}\sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)}\Big)^{-1} \\ &- \sum_{\tau\in\boldsymbol{\tau}_m(\bar{T})} \frac{1}{\beta}(1-e^{-\beta(\bar{T}-\tau)}) \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial \alpha_m^2} = &-\sum_{n=1}^{N(\bar{T})} \Big(\mathbb{1}_{\{\boldsymbol{\tau}_m(t_n)\neq\emptyset\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} e^{-\beta_m(t_n-\tau)}\Big)^2 \\ &\times\Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n)=i, X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2}\sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)}\Big)^{-2} \\ &< 0 \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial \alpha_1 \partial \alpha_2} = &-\sum_{n=1}^{N(\bar{T})} \prod_{m=1,2} \Big(\mathbb{1}_{\{\boldsymbol{\tau}_m(t_n)\neq\emptyset\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} e^{-\beta_m(t_n-\tau)}\Big) \\ &\times\Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n)=i, X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2}\sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)}\Big)^{-2} \\ &< 0 \end{aligned}$$

$$
\begin{aligned}
\frac{\partial l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial q_{ij}} &= \sum_{n=1}^{N(T)} \mathbb{1}_{\{X(t_n=i,X(t_n^+)=j\}} \\
&\times \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n=i,X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)} \Big)^{-1} \\
&- \int_0^{\bar{T}} \sum_{i=1}^{h-1} \mathbb{1}_{\{X(x)=i\}} dx
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial q_{ij}^2} &= -\sum_{n=1}^{N(T)} \mathbb{1}_{\{X(t_n=i,X(t_n^+)=j\}} \\
&\times \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n=i,X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)} \Big)^{-2} \\
&< 0
\end{aligned}
$$

$$
\frac{\partial^2 l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial q_{ij}\partial q_{kl}}\Big|_{(i,j)\neq(k,l)} = \frac{\partial^2 l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial q_{kl}\partial q_{ij}}\Big|_{(i,j)\neq(k,l)} = 0
$$

$$
\begin{aligned}
\frac{\partial l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial \beta_m} &= -\sum_{n=1}^{N(\bar{T})} \mathbb{1}_{\{\boldsymbol{\tau}_m(t_n)\neq\emptyset\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m(t_n-\tau)e^{-\beta_m(t_n-\tau)} \\
&\times \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n)=i,X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)} \Big)^{-1} \\
&+ \sum_{\tau\in\boldsymbol{\tau}_m(\bar{T})} \frac{\alpha_m}{\beta_m^2}(1-e^{-\beta_m(\bar{T}-\tau)}) \\
&- \sum_{\tau\in\boldsymbol{\tau}_m(\bar{T})} \frac{\alpha_m}{\beta_m}(\bar{T}-\tau)e^{-\beta_m(\bar{T}-\tau)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial \beta_m^2} &= -\sum_{n=1}^{N(\bar{T})} \Big(\mathbb{1}_{\{\boldsymbol{\tau}_m(t_n)\neq\emptyset\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m(t_n-\tau)e^{-\beta_m(t_n-\tau)} \Big)^2 \\
&\times \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n)=i,X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)} \Big)^{-2} \\
&+ \sum_{n=1}^{N(\bar{T})} \mathbb{1}_{\{\boldsymbol{\tau}_m(t_n)\neq\emptyset\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m(t_n-\tau)^2 e^{-\beta_m(t_n-\tau)} \\
&\times \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n)=i,X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)} \Big)^{-1} \\
&- 2\sum_{\tau\in\boldsymbol{\tau}_m(\bar{T})} \frac{\alpha_m}{\beta_m^3}(1-e^{-\beta_m(\bar{T}-\tau)}) \\
&+ 2\sum_{\tau\in\boldsymbol{\tau}_m(\bar{T})} \frac{\alpha_m}{\beta_m^2}(\bar{T}-\tau)e^{-\beta_m(\bar{T}-\tau)} \\
&+ \sum_{\tau\in\boldsymbol{\tau}_m(\bar{T})} \frac{\alpha_m}{\beta_m}(\bar{T}-\tau)^2 e^{-\beta_m(\bar{T}-\tau)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial \beta_1\partial\beta_2} &= -\sum_{n=1}^{N(\bar{T})} \prod_{m=1,2} \mathbb{1}_{\{\boldsymbol{\tau}_m(t_n)\neq\emptyset\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m(t_n-\tau)e^{-\beta_m(t_n-\tau)} \\
&\times \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n)=i,X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)} \Big)^{-2}
\end{aligned}
$$

$$\frac{\partial l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial \alpha_m \partial q_{ij}} = \frac{\partial l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial q_{ij}\partial \alpha_m}$$

$$= -\sum_{n=1}^{N(T)} \mathbb{1}_{\{X(t_n=i,X(t_n^+)=j\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} e^{-\beta_m(t_n-\tau)}$$

$$\times \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n=i,X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2}\sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)}\Big)^{-2}$$

$$\frac{\partial l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial \beta_m \partial q_{ij}} = \frac{\partial l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial q_{ij}\partial \beta_m}$$

$$= \sum_{n=1}^{N(T)} \mathbb{1}_{\{X(t_n=i,X(t_n^+)=j\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m (t_n-\tau) e^{-\beta_m(t_n-\tau)}$$

$$\times \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n=i,X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2}\sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)}\Big)^{-2}$$

$$\frac{\partial l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial \alpha_m \partial \beta_m} = \frac{\partial l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial \beta_m\partial \alpha_m}$$

$$= -\sum_{n=1}^{N(\bar{T})} \mathbb{1}_{\{\boldsymbol{\tau}_m(t_n)\neq\emptyset\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} (t_n-\tau) e^{-\beta_m(t_n-\tau)}$$

$$\times \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n)=i,X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2}\sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)}\Big)^{-1}$$

$$+ \sum_{n=1}^{N(\bar{T})} \Big(\mathbb{1}_{\{\boldsymbol{\tau}_m(t_n)\neq\emptyset\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} e^{-\beta_m(t_n-\tau)}\Big)$$

$$\times \Big(\mathbb{1}_{\{\boldsymbol{\tau}_m(t_n)\neq\emptyset\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m (t_n-\tau) e^{-\beta_m(t_n-\tau)}\Big)$$

$$\times \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n)=i,X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2}\sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)}\Big)^{-2}$$

$$+ \sum_{\tau\in\boldsymbol{\tau}_m(\bar{T})} \frac{1}{\beta_m^2}(1-e^{-\beta_m(\bar{T}-\tau)})$$

$$- \sum_{\tau\in\boldsymbol{\tau}_m(\bar{T})} \frac{1}{\beta_m}(\bar{T}-\tau) e^{-\beta_m(\bar{T}-\tau)})$$

$$\frac{\partial l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial \alpha_{m_1} \partial \beta_{m_2}}\Big|_{m_1\neq m_2} = \frac{\partial l(\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta}|\boldsymbol{X})}{\partial \beta_{m_2}\partial \alpha_{m_1}}\Big|_{m_1\neq m_2}$$

$$= \sum_{n=1}^{N(\bar{T})} \Big(\mathbb{1}_{\{\boldsymbol{\tau}_{m_1}(t_n)\neq\emptyset\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_{m_1}(t_n)} e^{-\beta_{m_1}(t_n-\tau)}\Big)$$

$$\times \Big(\mathbb{1}_{\{\boldsymbol{\tau}_{m_2}(t_n)\neq\emptyset\}} \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{\tau\in\boldsymbol{\tau}_{m_2}(t_n)} \alpha_{m_2} (t_n-\tau) e^{-\beta_{m_2}(t_n-\tau)}\Big)$$

$$\times \Big(\sum_{i,j=1}^{h} \mathbb{1}_{\{X(t_n)=i,X(t_n^+)=j\}} q_{ij} + \frac{\mathbb{1}_{X(t_n^+)>X(t_n))}}{N_i} \sum_{m=1,2}\sum_{\tau\in\boldsymbol{\tau}_m(t_n)} \alpha_m e^{-\beta_m(t_n-\tau)}\Big)^{-2}$$

5.2.4.4 The delta method

Let $\boldsymbol{Q} \in \mathbb{R}^{3\times 3}$ with $q_{ij} = q \in \mathbb{R}^+$ for any $i, j \in \{1, 2, 3\}$ with $i \neq j$ and $1 \neq 3$. To consider the probability of $p_{13}(T)$ per possible path the realization with $n = 1$ transition $\{1, 3\}$ is looked at. For any embedded paths from state 1 at time 0 to state 3 at time T the number of possible realizations is 1 for every value of $n \in \mathbb{N}^+$. The probability $\mathbb{P}(\{1,3\})$ is defined as follows with t_n transition times for $n = 1, 2, 3, ...$

$$\begin{aligned}\int_0^T 2qe^{-2qt_1}\frac{q}{2q}dt_1 &= q\int_0^T e^{-2qt_1}dt_1\\ &= \frac{1}{2}(1-e^{-2qT})\end{aligned}$$

now for $n = 2$ the embedded rating chain $\{1, 2, 3\}$

$$\begin{aligned}\int_0^T\int_0^{t_2} 2qe^{-2qt_1}\frac{q}{2q}2qe^{-2q(t_2-t_1)}\frac{q}{2q}dt_1dt_2 &= q^2\int_0^T\int_0^{t_2} e^{-2qt_2}dt_1dt_2\\ &= q^2\int_0^T t_2e^{-2qt_2}dt_2\end{aligned}$$

now for $n = 3$ the embedded rating chain $\{1, 2, 1, 3\}$

$$\begin{aligned}\int_0^T\int_0^{t_3}\int_0^{t_2} 2qe^{-2qt_1}\frac{q}{2q}2qe^{-2q(t_2-t_1)}\frac{q}{2q}2qe^{-2q(t_3-t_2)}\frac{q}{2q}dt_1dt_2dt_3 &= q^3\int_0^T\int_0^{t_3}\int_0^{t_2} e^{-2qt_3}dt_1dt_2dt_3\\ &= q^3\int_0^T \frac{t_3^2}{2}e^{-2qt_3}dt_3\end{aligned}$$

To conclude that $p_{13}(T)$ is defined by the sum of the probability of the embedded rating chains with n transitions from state 1 at time 0 to state 3 at time T

$$p_{13}(T) = \sum_{n=1}^{\infty} q^n \int_0^T \frac{x^{n-1}}{(n-1)!}e^{-2qx}dx$$

after which the order of integration and taking derivatives according to Leibniz rule could be applied to use the closed form expression for $p_{13}(t)$ in the delta method.

5.2.4.5 Modified thinning simulation

Let $N(t)$ be a time homogeneous Poisson process over time horizon $[0, T]$ with intensity function $\bar{\lambda}$ and $N = N(T)$ event times $t_1, t_2, \cdots, t_N$. Consider a time-inhomogenuous intensity function as follows $0 \leq \lambda(t) \leq \bar{\lambda}$ for any $t \in [0, T]$. Thin for $n = 1, 2, ..., N$ the event times t_N with probability $1 - \lambda(t_n)/\bar{\lambda}$, then the remaining event times follow a time-inhomogenuous Poisson process with intensity $\lambda(t)$ as follows [46] [13] [19]

$$\begin{aligned}\mathbb{P}(N(T) = N) &= \int_0^T\int_{t_1}^T\cdots\int_{t_{N-1}}^T \bar{\lambda}^n e^{-\bar{\lambda}T}\prod_{n=1}^{N}\Big(1-\frac{\lambda(t_n)}{\bar{\lambda}}\Big)dt_N\cdots dt_2dt_1\\ &=\frac{e^{-\bar{\lambda}}T}{N!}\int_0^T\int_0^T\int_0^T\prod_{n=1}^{N}(\bar{\lambda}-\lambda(t_n))ds_1ds_2\cdots ds_n\\ &=\frac{e^{-\bar{\lambda}}T}{N!}\Big(\int_0^T(\bar{\lambda}-\lambda(t_n))ds\Big)^N\\ &=\frac{e^{-\bar{\lambda}}T}{N!}\Big(\bar{\lambda}T-\int_a^b\lambda(s)ds\Big)^N\end{aligned}$$

with $\{s_1, s_2, ..., s_N\}$ the unordered event times and the summed probability of all possible number of events N should retrieve the time inhomogenuous Poisson process driven by $\lambda(t)$ as follows [46] [13] [19]

$$\begin{aligned}\sum_{n=0}^{\infty}\mathbb{P}(N(T) = n) &= e^{-\bar{\lambda}T}\sum_{n=0}^{\infty}\Big(\bar{\lambda}T-\int_0^T\lambda(s)ds\Big)^n/n!\\ &= e^{-\bar{\lambda}T}\cdot e^{\bar{\lambda}T-\int_0^T\lambda(s)ds}\\ &= e^{-\int_0^T\lambda(s)ds}\end{aligned}$$

5.2.5 Alternative methodologies

5.2.5.1 Time-inhomogeneous Chapman-Kolmogorov equations

Assuming the Markovian property for $\boldsymbol{X}(t)$ driven by $\boldsymbol{P}$, the Chapman-Kolmogorov equations for any $i, j \in S$ and $t_1, t_2, t_3 \in [0, T]$ such that $t_1 \leq t_2 \leq t_3$ are defined by [24]

$$(\boldsymbol{P}(t_1, t_3))_{ij} = \sum_{k \in S} (\boldsymbol{P}(t_1, t_2))_{ik} (\boldsymbol{P}(t_2, t_3))_{jk}$$

and given an infinitesmal timestep Δt and generator matrix $\boldsymbol{Q}$, the forward (and backward) equations are derived for any $t, t + \Delta t \in [0, T]$ as follows [24]

$$\begin{aligned}
(\boldsymbol{P})_{ij}(t + \Delta t) &= \sum_{k \in S} (\boldsymbol{P})_{ik}(t) (\boldsymbol{P})_{kj}(\Delta t) && \text{Chapman-Kolmogorov} \\
&= (\boldsymbol{P})_{ij}(t) (\boldsymbol{P})_{jj}(\Delta t) + \sum_{k \in S, k \neq j} (\boldsymbol{P})_{ik}(t) (\boldsymbol{P})_{kj}(\Delta t) && \text{Rearranged} \\
&\approx (\boldsymbol{P})_{ij}(t) (1 - q_{jj}(\Delta t) + \sum_{k \in S, k \neq j} (\boldsymbol{P})_{ik}(t) q_{kj} \Delta t && \text{Asymptotically} \\
&= (\boldsymbol{P})_{ij}(t) + (\boldsymbol{P})_{ij}(t) q_{jj}(\Delta t) + \sum_{k \in S, k \neq j} (\boldsymbol{P})_{ik}(t) q_{kj} \Delta t && \text{Rearranged} \\
&= (\boldsymbol{P})_{ij}(t) + \sum_{k \in S} (\boldsymbol{P})_{ik}(t) q_{kj} \Delta t && \text{-}
\end{aligned}$$

with

$$\frac{\boldsymbol{P}_{ij}(t + \Delta t) - \boldsymbol{P}_{ij}(t)}{\Delta t} = \boldsymbol{P}'(t) \approx \sum_{k \in S} (\boldsymbol{P})_{ik}(t) q_{kj}$$

and thus the forward equation in general matrix form is $\boldsymbol{P}'(t) = \boldsymbol{P}(t)\boldsymbol{Q}$ (and backward $\boldsymbol{P}'(t) = \boldsymbol{Q}\boldsymbol{P}(t)$).

As time-homogeneity is not necessarily assumed for $\boldsymbol{X}(t)$, the forward (and backward) equation holds for time-inhomogeneous $\boldsymbol{Q}$ as follows [24]

$$\boldsymbol{P}'(t) = \boldsymbol{P}(t)\boldsymbol{Q}(t)$$

and backward equation

$$\boldsymbol{P}'(t) = \boldsymbol{Q}(t)\boldsymbol{P}(t)$$

5.2.5.2 Discretized simulation

Let $\boldsymbol{Q} \in \mathbb{R}^{3 \times 3}$, $\alpha, \beta, T \in \mathbb{R}$ and $N \in \mathbb{N}^+$ with $\Delta t = T/N$. Hypothesize $(\boldsymbol{P}(n \cdot \Delta t))_{13} = ((\mathbf{I}_3 + \boldsymbol{Q}\Delta t)^n)_{13} + (\sum_i^n \boldsymbol{\Phi}_i)_{13}$ for some $\boldsymbol{\Phi}_n \in \mathbb{R}^{3 \times 3}$ by making no distinction between investment- or speculative grades in the non-Markovian model, such that $\boldsymbol{\alpha} = \alpha$ and $\boldsymbol{\beta} = \beta$. According a Maclaurin series with $f(x) = e^x$ for small enough Δt and intensity $\lambda(t)_i$ an approximation for $\boldsymbol{P}$ is as follows

$$(\boldsymbol{P}(t))_{ii} \approx 1 - (\lambda_i(t))\Delta t$$

and for any $i, j \in$ such that $j \neq i$

$$(\boldsymbol{P}(t))_{ij} \approx \lambda_i(t) f(i \to j | t) \Delta t$$

An attempt to simulate $(\boldsymbol{P})_{13}$ by discretizing $[0, T]$ such that $t_0 = 0, t_1 = \Delta t, t_2 = 2\Delta t, ..., t_N = T$ is as follows

Input: $\boldsymbol{Q}, \alpha, \beta, T, N$

Initialize $\boldsymbol{\Psi}_1 = \emptyset, \boldsymbol{\Phi}_1 = \emptyset, \boldsymbol{\Psi}_2 = \hat{\boldsymbol{Q}}_{1,d+u} \cdot e^{-\beta\Delta t}, \boldsymbol{\Phi}_2 = \hat{\boldsymbol{Q}}_{1,d} \cdot e^{-\beta\Delta t} \otimes \mathbf{N}_{\alpha,\Delta t}$;

Set $\Delta t = T/N$

for $n = 3, ...N$

  Set $\boldsymbol{Q}_n = \boldsymbol{Q}_{n-1} \times (\hat{\boldsymbol{Q}} e^{-\beta\Delta t} + \hat{\boldsymbol{Q}}_d)$

  Set $\boldsymbol{\Phi}_n = \boldsymbol{Q}_n \otimes \mathbf{N}_{\alpha,\Delta t} +$

   $(\hat{\boldsymbol{Q}}_{1,u} \times \hat{\boldsymbol{Q}}_u^{n-3} \times \hat{\boldsymbol{Q}}_d) \cdot e^{-\beta\Delta t} \otimes \mathbf{N}_{\alpha,\Delta t} +$

   $\boldsymbol{\Phi}_{n-1} \times \hat{\boldsymbol{Q}} +$

$$\mathbf{\Phi}_{n-1} \cdot e^{-(n-1)\beta\Delta t} \otimes N_{\alpha,\Delta t}$$

with

$$\hat{\boldsymbol{Q}} = \begin{pmatrix} 1-q_1\Delta t & q_{12}\Delta t & q_{13}\Delta t \\ q_{21}\Delta t & 1-q_2\Delta t & q_{23}\Delta t \\ 0 & 0 & 0 \end{pmatrix}$$

and

$$\hat{\boldsymbol{Q}}_u = \begin{pmatrix} 1-q_1\Delta t & 0 & 0 \\ q_{21}\Delta t & 1-q_2\Delta t & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

and

$$\hat{\boldsymbol{Q}}_{1,u} = \begin{pmatrix} 1-q_1\Delta t & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

and

$$\hat{\boldsymbol{Q}}_d = \begin{pmatrix} 0 & q_{12}\Delta t & q_{13}\Delta t \\ 0 & 0 & q_{23}\Delta t \\ 0 & 0 & 0 \end{pmatrix}$$

and

$$\hat{\boldsymbol{Q}}_{1,d} = \begin{pmatrix} 0 & q_{12}\Delta t & q_{13}\Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

and for any $\mathbf{M} \in \mathbb{R}^{3\times 3}$

$$\mathbf{M} \otimes \mathbf{N}_{\alpha,\Delta t} = \begin{pmatrix} -\Upsilon_1 & \frac{\Upsilon_1}{2} - \Upsilon_2 & \frac{\Upsilon_1}{2} + \Upsilon_2 \\ 0 & \frac{\Upsilon_1}{2} - \Upsilon_2 & \frac{\Upsilon_1}{2} + \Upsilon_2 \\ 0 & 0 & 0 \end{pmatrix}$$

for any $j \in \{1, 2\}$

$$\Upsilon_j = \alpha\Delta t \sum_{i=1}^{3} (\mathbf{M})_{ij}$$

with

$$(\mathbf{\Phi}_1)_{13} = 0$$

and

$$(\mathbf{\Phi}_2)_{13} = (q_{12}\Delta t)(\alpha e^{-\beta\Delta t}\Delta t)$$

and

$$\begin{aligned}(\mathbf{\Phi}_3)_{13} &= (q_{12}\Delta t)(q_{21}\Delta t)(\frac{\alpha}{2}e^{-2\beta\Delta t}\Delta t) \\ &+ (q_{12}\Delta t)(1-q_2\Delta t)(\alpha e^{-2\beta\Delta t}\Delta t) \\ &+ (1-q_1\Delta t)(q_{12}\Delta t)(\alpha e^{-\beta\Delta t}\Delta t) \\ &+ (q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(q_{23}\Delta t) \\ &+ (q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)\end{aligned}$$

and

$$
\begin{aligned}
(\mathbf{\Phi}_4)_{13} = & (q_{12}\Delta t)(q_{21}\Delta t)(1-q_1\Delta t)(\frac{\alpha}{2}e^{-3\beta\Delta t}\Delta t) \\
&+ (q_{12}\Delta t)(q_{21}\Delta t)(q_{12}\Delta t)(\alpha e^{-3\beta\Delta t}\Delta t) \\
&+ (q_{12}\Delta t)(1-q_2\Delta t)(q_{21}\Delta t)(\frac{\alpha}{2}e^{-3\beta\Delta t}\Delta t) \\
&+ (q_{12}\Delta t)(1-q_2\Delta t)(1-q_2\Delta t)(\alpha e^{-3\beta\Delta t}\Delta t) \\
&+ (q_{12}\Delta t)(q_{21}\Delta t)(q_{12}\Delta t)(e^{-2\beta\Delta t}\Delta t) \\
&+ (1-q_1\Delta t)(1-q_1\Delta t)(q_{12}\Delta t)(\alpha e^{-\beta\Delta t}\Delta t) \\
&+ (q_{12}\Delta t)(1-q_1\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)(q_{23}\Delta t) \\
&+ (q_{12}\Delta t)(q_{21}\Delta t)(\frac{\alpha}{2}e^{-2\beta\Delta t}\Delta t)(q_{23}\Delta t) \\
&+ (q_{12}\Delta t)(q_{21}\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)(q_{13}\Delta t) \\
&+ (1-q_1\Delta t)(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(q_{23}\Delta t) \\
&+ (q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(1-q_2\Delta t)(q_{23}\Delta t) \\
&+ (q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)(q_{23}\Delta t) \\
&+ (q_{12}\Delta t)(1-q_2\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)(\alpha e^{-3\beta\Delta t}\Delta t) \\
&+ (q_{12}\Delta t)(q_{21}\Delta t)(\frac{\alpha}{2}e^{-2\beta\Delta t}\Delta t)(\alpha e^{-3\beta\Delta t}\Delta t) \\
&+ (q_{12}\Delta t)(q_{21}\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)(\frac{\alpha}{2}e^{-3\beta\Delta t}\Delta t) \\
&+ (1-q_1\Delta t)(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(\alpha e^{-3\beta\Delta t}\Delta t) \\
&+ (q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(1-q_2\Delta t)(\alpha e^{-3\beta\Delta t}\Delta t) \\
&+ (q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)(\alpha e^{-3\beta\Delta t}\Delta t) \\
&+ (q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(q_{12}\Delta t)(\frac{\alpha}{2}e^{-3\beta\Delta t}\Delta t)
\end{aligned}
$$

which does not capture $\{1,2,1,2,3\}$ properly.

Unfortunately, for $n = 4$, $(\sum_i^n \mathbf{\Phi}_i)_{13}$ does not capture the non-Markovian momentum contribution properly as demonstrated below. This might imply an iterative scheme is not possible and all possible realizations have to simulated separately implying too much computational requirements for a long time horizon $[0, T]$.

n	$\Omega \in \mathcal{F}_n$	$\mathbb{P}(\Omega)$	$(\boldsymbol{P}^n_{\Delta t})_{13}$	$(\boldsymbol{\Psi}_n)_{13}$
1	$\{1,3\}$	$(q_{13}\Delta t)$	$(q_{13}\Delta t)$	0
2	$\{1,1,3\}$	$(1-q_1\Delta t)(q_{13}\Delta t)$	$(1-q_1\Delta t)(q_{13}\Delta t)$	0
	$\{1,2,3\}$	$(q_{12}\Delta t)((q_{23}+\alpha e^{-\beta\Delta t})\Delta t)$	$(q_{12}\Delta t)(q_{23}\Delta t)$	$(q_{12}\Delta t)(\alpha e^{-\beta\Delta t}\Delta t)$
	$\{1,3,3\}$	$(q_{13}\Delta t)$	$(q_{13}\Delta t)$	0
3	$\{1,1,1,3\}$	$(1-q_1\Delta t)(1-q_1\Delta t)(q_{13}\Delta t)$	$(1-q_1\Delta t)(1-q_1\Delta t)(q_{13}\Delta t)$	0
	$\{1,1,2,3\}$	$(1-q_1\Delta t)(q_{12}\Delta t)(q_{23}+\alpha e^{-\beta\Delta t})\Delta t)$	$(1-q_1\Delta t)(q_{12}\Delta t)(q_{23}\Delta t)$	$(1-q_1\Delta t)(q_{12}\Delta t)(\alpha e^{-\beta\Delta t}\Delta t)$
	$\{1,1,3,3\}$	$(1-q_1\Delta t)(q_{13}\Delta t)$	$(1-q_1\Delta t)(q_{13}\Delta t)$	0
	$\{1,2,1,3\}$	$(q_{12}\Delta t)(q_{21}\Delta t)((q_{13}+\alpha/2e^{-2\beta\Delta t})\Delta t)$	$(q_{12}\Delta t)((q_{21}\Delta t)(q_{13}\Delta t)$	$(q_{12}\Delta t)(q_{21}\Delta t)(\alpha/2e^{-2\beta\Delta t}\Delta t)$
	$\{1,2,2,3\}$	$(q_{12}\Delta t)(1-(q_2+\alpha e^{-\beta\Delta t})\Delta t)((q_{23}+\alpha e^{-2\beta\Delta t})\Delta t)$	$(q_{12}\Delta t)(1-q_2\Delta t)(q_{23}\Delta t)$	$(q_{12}\Delta t)(1-q_2\Delta t)(\alpha e^{-2\beta\Delta t}\Delta t)+$ $(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(q_{23}\Delta t)+$ $(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(\alpha e^{-2\beta\Delta t}\Delta t)$
	$\{1,2,3,3\}$	$(q_{12}\Delta t)((q_{23}+\alpha e^{-\beta\Delta t})\Delta t)$	$(q_{12}\Delta t)(q_{23}\Delta t)$	$(q_{12}\Delta t)(\alpha e^{-\beta\Delta t}\Delta t)$
	$\{1,3,3,3\}$	$(q_{13}\Delta t)$	$(q_{13}\Delta t)$	0
4	$\{1,1,1,1,3\}$	$(1-q_1\Delta t)(1-q_1\Delta t)(1-q_1\Delta t)(q_{13}\Delta t)$	$(1-q_1\Delta t)(1-q_1\Delta t)(1-q_1\Delta t)(q_{13}\Delta t)$	0
	$\{1,1,1,2,3\}$	$(1-q_1\Delta t)(1-q_1\Delta t)(q_{12}\Delta t)((q_{23}+\alpha e^{-\beta\Delta t})\Delta t)$	$(1-q_1\Delta t)(1-q_1\Delta t)(q_{12}\Delta t)(q_{23}\Delta t)$	$(1-q_1\Delta t)(1-q_1\Delta t)(q_{12}\Delta t)(\alpha e^{-\beta\Delta t}\Delta t)$
	$\{1,1,1,3,3\}$	$(1-q_1\Delta t)(1-q_1\Delta t)(q_{13}\Delta t)$	$(1-q_1\Delta t)(1-q_1\Delta t)(q_{13}\Delta t)$	0
	$\{1,1,2,1,3\}$	$(1-q_1\Delta t)(q_{12}\Delta t)(q_{21}\Delta t)((q_{13}+\alpha/2e^{-2\beta\Delta t})\Delta t)$	$(1-q_1\Delta t)(q_{12}\Delta t)(q_{21}\Delta t)(q_{13}\Delta t)$	$(1-q_1\Delta t)(q_{12}\Delta t)(q_{21}\Delta t)(\alpha/2e^{-2\beta\Delta t}\Delta t)$
	$\{1,1,2,2,3\}$	$(1-q_1\Delta t)(q_{12}\Delta t)(1-(q_2+\alpha e^{-\beta\Delta t})\Delta t)((q_{23}+\alpha e^{-2\beta\Delta t})\Delta t)$	$(1-q_1\Delta t)(q_{12}\Delta t)(1-q_2\Delta t)(q_{23}\Delta t)$	$(1-q_1\Delta t)(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t))(q_{23}\Delta t)$ $(1-q_1\Delta t)(q_{12}\Delta t)(1-q_2\Delta t)(\alpha e^{-2\beta\Delta t}\Delta t)$ $(1-q_1\Delta t)(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(\alpha e^{-2\beta\Delta t}\Delta t)$
	$\{1,1,2,3,3\}$	$(1-q_1\Delta t)(q_{12}\Delta t)((q_{23}+\alpha e^{-\beta\Delta t})\Delta t)$	$(1-q_1\Delta t)(q_{12}\Delta t)(q_{23}\Delta t)$	$(1-q_1\Delta t)(q_{12}\Delta t)(\alpha e^{-\beta\Delta t}\Delta t)$
	$\{1,1,3,3,3\}$	$(1-q_1\Delta t)(q_{13}\Delta t)$	$(1-q_1\Delta t)(q_{13}\Delta t)$	0
	$\{1,2,1,1,3\}$	$(q_{12}\Delta t)(q_{21}\Delta t)(1-(q_1+\alpha e^{-2\beta\Delta t})\Delta t)((q_{13}+\alpha/2e^{-3\beta\Delta t})\Delta t)$	$(q_{12}\Delta t)(q_{21}\Delta t)(1-q_1\Delta t)(q_{13}\Delta t)$	$(q_{12}\Delta t)(q_{21}\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)(q_{13}\Delta t)$ $(q_{12}\Delta t)(q_{21}\Delta t)(1-q_1\Delta t)(\alpha/2e^{-3\beta\Delta t}\Delta t)$ $(q_{12}\Delta t)(q_{21}\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)(\alpha/2e^{-3\beta\Delta t}\Delta t)$
	$\{1,2,1,2,3\}$	$(q_{12}\Delta t)(q_{21}\Delta t)((q_{12}+\alpha/2e^{-2\beta\Delta t})\Delta t)((q_{23}+\alpha e^{-3\beta\Delta t}+\alpha e^{-\beta\Delta t})\Delta t)$	$(q_{12}\Delta t)(q_{21}\Delta t)(q_{12}\Delta t)(q_{23}\Delta t)$	$(q_{12}\Delta t)(q_{21}\Delta t)(\alpha/2e^{-2\beta\Delta t}\Delta t)(q_{23}\Delta t)$ $(q_{12}\Delta t)(q_{21}\Delta t)(q_{12}\Delta t)((\alpha e^{-3\beta\Delta t}+\alpha e^{-\beta\Delta t})\Delta t)$ $(q_{12}\Delta t)(q_{21}\Delta t)(\alpha/2e^{-2\beta\Delta t}\Delta t)((\alpha e^{-3\beta\Delta t}+\alpha e^{-\beta\Delta t})\Delta t)$
	$\{1,2,1,3,3\}$	$(q_{12}\Delta t)(q_{21}\Delta t)((q_{13}+\alpha/2e^{-2\beta\Delta t})\Delta t)$	$(q_{12}\Delta t)(q_{21}\Delta t)(q_{13}\Delta t)$	$(q_{12}\Delta t)(q_{21}\Delta t)(\alpha/2e^{-2\beta\Delta t}\Delta t)$
	$\{1,2,2,1,3\}$	$(q_{12}\Delta t)(1-(q_2+\alpha e^{-\beta\Delta t})\Delta t)(q_{21}\Delta t)((q_{13}+\alpha/2e^{-3\beta\Delta t})\Delta t)$	$(q_{12}\Delta t)(1-q_2\Delta t)(q_{21}\Delta t)(q_{13}\Delta t)$	$(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(q_{21}\Delta t)(q_{13}\Delta t)$ $(q_{12}\Delta t)(1-q_2\Delta t)(q_{21}\Delta t)(\alpha/2e^{-3\beta\Delta t}\Delta t)$ $(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(q_{21}\Delta t)(\alpha/2e^{-3\beta\Delta t}\Delta t)$
	$\{1,2,2,2,3\}$	$(q_{12}\Delta t)(1-(q_2+\alpha e^{-\beta\Delta t})\Delta t)(1-(q_2+\alpha e^{-2\beta\Delta t})\Delta t)((q_{23}+\alpha e^{-3\beta\Delta t})\Delta t)$	$(q_{12}\Delta t)(1-q_2\Delta t)(1-q_2\Delta t)(q_{23}\Delta t)$	$(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(1-q_2\Delta t)(q_{23}\Delta t)+$ $(q_{12}\Delta t)(1-q_2\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)(q_{23}\Delta t)+$ $(q_{12}\Delta t)(1-q_2\Delta t)(1-q_2\Delta t)(\alpha e^{-3\beta\Delta t}\Delta t)+$ $(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)(q_{23}\Delta t)+$ $(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(1-q_2\Delta t)(\alpha e^{-3\beta\Delta t}\Delta t)+$ $(q_{12}\Delta t)(1-q_2\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)(\alpha e^{-3\beta\Delta t}\Delta t)+$ $(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(-\alpha e^{-2\beta\Delta t}\Delta t)(\alpha e^{-3\beta\Delta t}\Delta t)$
	$\{1,2,2,3,3\}$	$(q_{12}\Delta t)(1-(q_2+\alpha e^{-\beta\Delta t})\Delta t)((q_{23}+\alpha e^{-2\beta})\Delta t)$	$(q_{12}\Delta t)(1-q_2\Delta t)(q_{23}\Delta t)$	$(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(q_{23}\Delta t)+$ $(q_{12}\Delta t)(1-q_2\Delta t)(\alpha e^{-2\beta}\Delta t)+$ $(q_{12}\Delta t)(-\alpha e^{-\beta\Delta t}\Delta t)(\alpha e^{-2\beta}\Delta t)$
	$\{1,2,3,3,3\}$	$(q_{12}\Delta t)((q_{23}+\alpha e^{-\beta\Delta t})\Delta t)$	$(q_{12}\Delta t)(q_{23}\Delta t)$	$(q_{12}\Delta t)(\alpha e^{-\beta\Delta t}\Delta t)$
	$\{1,3,3,3,3\}$	$(q_{13}\Delta t)$	$(q_{13}\Delta t)$	0

5.3 Relevant Matlab scripts

5.3.1 Modified thinning simulation

```
%% Modified thinning simulation (~5 seconds)
clc
clear
close all
tic
%% Model input
Q = ...
[-0.0869  0.0836  0.0031  0.0000  0.0002  0.0000  0.0000  0.0000 0.0000;
  0.0117 -0.1088  0.0942  0.0025  0.0003  0.0001  0.0000  0.0000 0.0000;
  0.0006  0.0240 -0.0938  0.0666  0.0017  0.0007  0.0002  0.0000 0.0000;
  0.0002  0.0016  0.0387 -0.0947  0.0496  0.0040  0.0006  0.0000 0.0000;
  0.0001  0.0006  0.0033  0.0636 -0.1774  0.1060  0.0037  0.0001 0.0000;
  0.0000  0.0003  0.0012  0.0035  0.0503 -0.1610  0.1012  0.0040 0.0004;
  0.0000  0.0002  0.0001  0.0013  0.0048  0.1028 -0.1976  0.0622 0.0261;
  0.0000  0.0000  0.0018  0.0029  0.0050  0.0447  0.1346 -0.2838 0.0948;
  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000 0.0000];
a = [0.0310 0.1291];
b = [3.5234 1.7095];
lig = 4;
lve = {'Aaa','Aa','A','Baa','Ba','B','Caa','Ca','D'};
%% Simulation input
T = 30;
N = [413 1313 2232 2318 2021 4504 1333 59];
%% Computations
dim = size(Q,1);
Nj = sum(triu(Q)>0,2);
Q(eye(dim)==1) = -sum(Q,2)+Q(eye(dim)==1);
M = sum(N);
%% Preallocations
[m,t1,t2,t3,t4,l1,l2,l3,l4,v1,v2] = deal(cell(M,1));
[I,J,K] = deal(ones(M,1));
[L,rig,rsg] = deal(zeros(M,1));
%% Modified thinning simulation
parfor i = 1:M
    m{i} = find(cumsum(N)>=i,1);
    mu = -Q(m{i},m{i});
    l1{i}(I(i)) = mu;
    l2{i}(I(i)) = mu;
    t1{i}(I(i)) = -log(rand)/l1{i};
    v1{i}(I(i)) = rand;
    l4{i}(I(i)) = l1{i}(I(i));
    P = cumsum([0 Q(m{i},1:m{i}-1) 0 Q(m{i},m{i}+1:dim)])./l2{i};
    I(i) = I(i)+1;
    m{i}(I(i)) = find(P>=rand,1)-1;
    mu = -Q(m{i}(I(i)),m{i}(I(i)));
    if m{i}(I(i)) > m{i}(I(i)-1)
        if m{i}(I(i)-1) <= lig
            t2{i} = t1{i};
            l1{i}(I(i)) = mu+a(1);
            t1{i}(I(i)) = t1{i}-log(rand)/l1{i}(I(i));
            rig(i) = exp(-b(1)*(t1{i}(I(i))-t1{i}(I(i)-1)));
            J(i) = J(i)+1;
        else
            t3{i} = t1{i};
            l1{i}(I(i)) = mu+a(2);
            t1{i}(I(i)) = t1{i}-log(rand)/l1{i}(I(i));
            rsg(i) = exp(-b(2)*(t1{i}(I(i))-t1{i}(I(i)-1)));
```

```matlab
            K(i) = K(i)+1;
        end
    else
        l1{i}(I(i)) = mu;
        t1{i}(I(i)) = t1{i}(I(i)-1)-log(rand)/l1{i}(I(i));
    end
    l4{i}(I(i)) = l1{i}(I(i));
    l2{i}(I(i)) = mu+a(1)*rig(i)+a(2)*rsg(i);
    while t1{i}(I(i)) < T && m{i}(I(i)) < dim
        v1{i}(I(i)) = rand;
        if v1{i}(I(i))*l1{i}(I(i)) <= l2{i}(I(i))
            dq = (l2{i}(I(i))-mu)/Nj(m{i}(I(i)));
            P = cumsum([0 Q(m{i}(I(i)),1:m{i}(I(i))-1) 0 ...
                Q(m{i}(I(i)),m{i}(I(i))+1:dim)+...
                (Q(m{i}(I(i)),m{i}(I(i))+1:dim)>0)*dq])./l2{i}(I(i));
            I(i) = I(i)+1;
            m{i}(I(i)) = find(P>=rand,1)-1;
            mu = -Q(m{i}(I(i)),m{i}(I(i)));
            if m{i}(I(i)) > m{i}(I(i)-1)
                if m{i}(I(i)-1) <= lig
                    t2{i}(J(i)) = t1{i}(I(i)-1);
                    l1{i}(I(i)) = dq*Nj(m{i}(I(i)-1))+mu+a(1);
                    t1{i}(I(i)) = t1{i}(I(i)-1)-log(rand)/l1{i}(I(i));
                    rig(i) = exp(-b(1)*(t1{i}(I(i))-t1{i}(I(i)-1)))*...
                        (1+rig(i));
                    rsg(i) = exp(-b(2)*(t1{i}(I(i))-t1{i}(I(i)-1)))*rsg(i);
                    J(i) = J(i)+1;
                else
                    t3{i}(K(i)) = t1{i}(I(i)-1);
                    l1{i}(I(i)) = dq*Nj(m{i}(I(i)-1))+mu+a(2);
                    t1{i}(I(i)) = t1{i}(I(i)-1)-log(rand)/l1{i}(I(i));
                    rig(i) = exp(-b(1)*(t1{i}(I(i))-t1{i}(I(i)-1)))*rig(i);
                    rsg(i) = exp(-b(2)*(t1{i}(I(i))-t1{i}(I(i)-1)))*...
                        (1+rsg(i));
                    K(i) = K(i)+1;
                end
            else
                l1{i}(I(i)) = dq*Nj(m{i}(I(i)-1))+mu;
                t1{i}(I(i)) = t1{i}(I(i)-1)-log(rand)/l1{i}(I(i));
                rig(i) = exp(-b(1)*(t1{i}(I(i))-t1{i}(I(i)-1)))*rig(i);
                rsg(i) = exp(-b(2)*(t1{i}(I(i))-t1{i}(I(i)-1)))*rsg(i);
            end
        else
            L(i) = L(i)+1;
            t4{i}(L(i)) = t1{i}(I(i));
            v2{i}(L(i)) = v1{i}(I(i));
            l3{i}(L(i)) = l1{i}(I(i));
            l1{i}(I(i)) = l2{i}(I(i));
            t1{i}(I(i)) = t1{i}(I(i))-log(rand)/l1{i}(I(i));
            rig(i) = exp(-b(1)*(t1{i}(I(i))-t4{i}(L(i))))*rig(i);
            rsg(i) = exp(-b(2)*(t1{i}(I(i))-t4{i}(L(i))))*rsg(i);
        end
        l4{i}(I(i)+L(i)) = l1{i}(I(i));
        l2{i}(I(i)) = mu+a(1)*rig(i)+a(2)*rsg(i);
    end
    t1{i} = [0 nonzeros(t1{i}(1:I(i)-1).*(t1{i}(1:I(i)-1)<T)).' T];
    m{i} = m{i}(1:length(t1{i})-1);
    I(i) = length(m{i});
    l1{i} = l1{i}(1:I(i));
    l2{i} = l2{i}(1:I(i));
    l1{i}(I(i)) = l1{i}(I(i))-l1{i}(I(i))*(m{i}(I(i))==dim(1));
```

```
    l2{i}(I(i)) = l2{i}(I(i))-l2{i}(I(i))*(m{i}(I(i))==dim(1));
    l4{i}(I(i)+L(i)) = l4{i}(I(i)+L(i))-l4{i}(I(i)+L(i))*...
        (m{i}(I(i))==dim(1));
end
%% Input
path = matlab.desktop.editor.getActiveFilename;
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '1. Modified thinning simulation input'));
save('Q','Q'); save('a','a'); save('b','b'); save('lig','lig');
save('T','T'); save('N','N'); save('lve','lve');
%% Output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '2. Modified thinning simulation output'));
save('dim','dim'); save('Nj','Nj'); save('v1','v1'); save('v2','v2');
save('t1','t1'); save('t2','t2'); save('t3','t3'); save('t4','t4');
save('l1','l1'); save('l2','l2'); save('l3','l3'); save('l4','l4');
save('I','I'); save('J','J'); save('K','K'); save('L','L'); save('m','m');
toc
```

5.3.2 Exact maximum likelihood estimation

```
%% Exact maximum likelihood estimator (~1 minute)
clc
clear
close all
tic
%% Modified thinning simulation input
path = matlab.desktop.editor.getActiveFilename;
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '1. Modified thinning simulation input'));
load('lig'); load('T'); load('N'); load('dt');
%% Modified thinning simulation output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '2. Modified thinning simulation output'));
load('m'); load('t1'); load('dim'); load('I'); load('Nj');
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2)));
%% Estimation input
x = 0.975;
%% Computations
O = round(T/dt);
M = sum(N);
%% Preallocations
Nij = cell(M,1);
Ri = cell(M,1);
%% Exact likelihood estimator
parfor i = 1:M
    Nij{i} = zeros(dim);
    Ri{i} = zeros(dim,1);
    for j = 1:dim-1
        Ri{i}(j) = sum((m{i}==j).*diff(t1{i}));
        for k = 1:dim
            Nij{i}(j,k) = sum((m{i}(1:I(i)-1)==j).*...
                (m{i}(2:I(i))==k));
        end
    end
end
mle = sum(cat(3,Nij{:}),3)./sum(cat(3,Ri{:}),3);
mle(eye(dim)==1) = -sum(mle,2)+mle(eye(dim)==1);
mle(isnan(mle)) = 0;
mle(dim,:) = 0;
%% Preallocations
v = cell(1,12);
C = cell(dim,dim);
dpv = cell(O+1,dim,dim);
pme = zeros(O+1,dim,dim);
pmi = zeros(O+1,dim-1,dim);
%% Computations
%dt = T/O;
[ver,hor,nrp,~,~] = fb_allowedpairsfunction(m,dim,1);
[t1,~,~,~,~,~,~,~,v6] = fc_vectorfunction(m,t1,ver,hor,Nj,lig);
v{1} = M; v{2} = nrp; v{3} = I; v{6} = cell(M,1); v{12} = v6;
h = fd_markovianloghessianfunction(m,mle,v);
f = -inv(h);
%% Confidence intervals
parfor i = 1:dim
    ei = zeros(dim,1); ei(i) = 1;
    for j = 1:dim
        ej = zeros(dim,1); ej(j) = 1;
        C{i,j} = [mle ei*ej.'-ei*ei.'; zeros(dim) mle];
    end
```

```
end
parfor i = 1:O+1
    t2 = (i-1)*dt;
    for j = 1:dim
        for k = 1:dim
            dpv{i,j,k} = zeros(1,nrp);
            for l = 1:nrp
                dum = expm(C{ver(l),hor(l)}*t2);
                dpv{i,j,k}(l) = dum(j,dim+k);
            end
        end
    end
end
%% Default probability
parfor i = 1:O+1
    pme(i,:,:) = expm(mle*(i-1)*dt);
end
for i = 1:O+1
    for j = 1:dim-1
        for k = 1:dim
            pmi(i,j,k) = norminv(x)*sqrt(dpv{i,j,k}*f*dpv{i,j,k}.');
        end
    end
end
%% Input
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '7. Exact maximum likelihood estimator input'));
save('x','x');
%% Output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '8. Exact maximum likelihood estimator output'));
save('mle','mle'); save('pme','pme'); save('pmi','pmi');
save('ver','ver'); save('hor','hor'); save('nrp','nrp');
toc
```

5.3.3 Expectation-maximization algorithm

```matlab
%% Expectation-maximization algorithm (~3 minutes)
close all
clc
clear
tic
%% Modified thinning simulation input
path = matlab.desktop.editor.getActiveFilename;
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '1. Modified thinning simulation input'));
load('lig'); load('T'); load('N'); load('dt');
%% Modified thinning simulation output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '2. Modified thinning simulation output'));
load('m'); t = load('t1'); t = t.t1; load('dim'); load('I');
%% Exact maximum likelihood estimator input
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '7. Exact maximum likelihood estimator input'));
load('x');
%% Exact maximum likelihood estimator output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '8. Exact maximum likelihood estimator output'));
load('mle'); load('ver'); load('hor'); load('nrp');
%% Estimation input
eps = 1e-9;
mit = 1e3;
%% Computations
mle = eps*(mle>eps);
O = round(T/dt);
M = sum(N);
%% Preallocations
dm1 = zeros(T+1,M);
ttm = zeros(T,dim,dim);
lnl = zeros(1,mit+1);
%% Computations
for h = 1:T+1
    for i = 1:M
        dm1(h,i) = m{i}(sum(find(t{i}(2:I(i))<h-1,1,'last'))+1);
        if h > 1
            ttm(h-1,dm1(h-1,i),dm1(h,i)) = ttm(h-1,dm1(h-1,i),dm1(h,i))+1;
        end
    end
end
%% Expectation maximization algorithm
i = 1;
while i < mit
    esj = zeros(1,dim);
    ekk = zeros(dim);
    for j = 1:dim
        ea = zeros(dim,1);
        ea(j) = 1;
        for l = 1:T
            ecp = expm([mle ea*ea.'; zeros(dim) mle]);
            eqt = expm(mle);
            esj(j) = esj(j)+...
                nansum(squeeze(ttm(l,:,:)).*ecp(1:dim,dim+1:2*dim)./...
                eqt,'all');
            for k = setdiff(linspace(1,dim,dim),j)
                eb = deal(zeros(dim,1));
                eb(k) = deal(1);
```

```
                ecg = expm([mle mle(j,k)*ea*eb.'; zeros(dim) mle]);
                ekk(j,k) = ekk(j,k)+...
                    nansum(squeeze(ttm(l,:,:)).*ecg(1:dim,dim+1:2*dim)./...
                    eqt,'all');
            end
        end
    end
    mle = ekk./esj.';
    mle(eye(dim)==1) = -sum(mle,2)+mle(eye(dim)==1);
    mle(isnan(mle)) = 0;
    for j = 1:dim
        for k = 1:dim
            for l = 1:T
                P = expm(mle);
                if P(j,k) > 0
                    lnl(i+1) = lnl(i+1)+ttm(l,j,k)*log(P(j,k));
                end
            end
        end
    end
    rer = abs(lnl(i+1)-lnl(i))/abs(lnl(i));
    if i > 2 && rer <= eps
        break
    end
    i = i+1;
end
lnl = lnl(lnl~=0);
%% Preallocations
C = cell(dim,dim);
pme = zeros(O+1,dim,dim);
pmi = zeros(O+1,dim-1,dim);
dpv = cell(O+1,dim,dim);
H = zeros(nrp);
%% Fisher information
for h = 1:nrp
    for i = 1:nrp
        dH = zeros(T,dim,dim);
        [a,b,m,v] = deal(ver(h),hor(h),ver(i),hor(i));
        [ea,eb,em,ev] = deal(zeros(dim,1));
        [ea(a),eb(b),em(m),ev(v)] = deal(1);
        ca = [mle ea*eb.'-ea*ea.'; zeros(dim) mle];
        cm = [mle em*ev.'-em*em.'; zeros(dim) mle];
        dc = [em*ev.'-em*em.' zeros(dim); zeros(dim) em*ev.'-em*em.'];
        cx = [ca dc; zeros(2*dim) ca];
        [eq,ecx] = deal(expm(mle),expm(cx));
        [eca,ecm] = deal(expm(ca),expm(cm));
        for j = 1:T
            for k = 1:dim
                for l = 1:dim
                    dH(j,k,l) = ttm(j,k,l)/eq(k,l)*...
                        (eca(k,dim+l)*ecm(k,dim+l)/eq(k,l)-ecx(k,3*dim+l));
                end
            end
        end
        H(h,i) = nansum(dH,'all');
    end
end
F = abs(inv(H));
%% Default probability & confidence intervals
for i = 1:dim
    ei = zeros(dim,1);
```

```
    ei(i) = 1;
    for j = 1:dim
        ej = zeros(dim,1);
        ej(j) = 1;
        C{i,j} = [mle ei*ej.'-ei*ei.'; zeros(dim) mle];
    end
end
for i = 1:O+1
    t2 = (i-1)*dt;
    for j = 1:dim
        for k = 1:dim
            dpv{i,j,k} = zeros(1,nrp);
            for l = 1:nrp
                dm2 = expm(C{ver(l),hor(l)}*t2);
                dpv{i,j,k}(l) = dm2(j,dim+k);
            end
        end
    end
end
parfor i = 2:O+1
    pme(i,:,:) = expm(mle*(i-1)*dt);
end
pme(1,:,:) = eye(dim);
for i = 1:O+1
    for j = 1:dim-1
        for k = 1:dim
            pmi(i,j,k) = norminv(x)*sqrt(dpv{i,j,k}*F*dpv{i,j,k}.');
        end
    end
end
%% Input
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '10. Expectation-maximization algorithm input'));
save('eps','eps'); save('mit','mit');
%% Output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '11. Expectation-maximization algorithm output'));
save('mle','mle'); save('pme','pme'); save('pmi','pmi'); save('lnl','lnl');
toc
```

5.3.4 Metropolis-Hastings algorithm

```
%% Metropolis-Hastings algortihm (~6 hours)
clc
clear
close all
set(groot,'defaulttextinterpreter','latex');
%% Estimation input
ite = 1e3;
bii = 1e2;
var = 1e-3;
%% Modified thinning simulation input
path = matlab.desktop.editor.getActiveFilename;
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '1. Modified thinning simulation input'));
load('a'); load('b'); load('lig'); load('N'); load('T');
%% Modified thinning simulation output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '2. Modified thinning simulation output'));
t = load('t1'); t = t.t1; load('m'); dm1 = load('dim'); dm1 = dm1.dim;
load('I');
%% Exact maximum likelihood estimator output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '8. Exact maximum likelihood estimator output'));
Q = load('mle'); Q = Q.mle; load('nrp');
%% Markovian projected Newton Raphson method output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '14. Markovian projected Newton-Raphson method output'));
load('v');
%% Non Markovian projected Newton Raphson method input
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '16. Non markovian projected Newton-Raphson method input'));
load('eps');
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2)));
%% Computations
M = sum(N);
ite = ite+bii;
dm2 = dm1^2;
a = exprnd(a);
b = exprnd(b);
mab = [[0.1; 1];[10; 1]];
%% Preallocations
x = zeros(dm2+4,1);
ip = zeros(dm2+4,1);
c = 0;
%% Computations
for i = 1:dm2
    if Q(i) > 0
        c = c+1;
        x(i) = Q(i);
        ip(c) = i;
    end
end
ip(dm2+1:dm2+4) = dm2+1:dm2+4;
ip(ip==0) = [];
x(dm2+1:dm2+4) = [a b];
%% Preallocations
x = [x repmat(zeros(dm2+4,1),1,ite)];
[r,l0,l1] = deal(zeros(nrp+4,ite+1));
%% Metropolis Hastings algorithm
P = waitbar(0,'\textbf{Please wait}','Name',...
```

```
    'Metropolis-Hastings algorithm');
s2 = 0;
l0(1,1) = fi_loglikelihoodfunction(m,t,Q,a,b,v);
i = 1;
while i <= ite
    tic
    for j = 1:nrp+4
        d = ip(j);
        while x(d,i+1) < eps
            x(d,i+1) = normrnd(x(d,i),var);
        end
        Q = reshape([x(1:min(d,dm2),i+1); x(d+1:dm2,i)],[dm1 dm1]);
        Q(eye(dm1)==1) = -sum(Q,2)+Q(eye(dm1)==1);
        a = [x(dm2+1:min(d,dm2+2),i+1).'; x((max(d+1,dm2+1):dm2+2),i)].';
        b = [x(dm2+3:min(d,dm2+4),i+1).'; x((max(d+1,dm2+3):dm2+4),i)].';
        l1(j,i) = fi_loglikelihoodfunction(m,t,Q,a,b,v);
        if ip(j) > dm2
            r(j,i) = exp(l1(j,i)-l0(j,i))*...
                exppdf(x(d,i+1),mab(ip(j)-dm2))/...
                exppdf(x(d,i),mab(ip(j)-dm2));
        else
            r(j,i) = exp(l1(j,i)-l0(j,i));
        end
        u = rand<=r(j,i);
        l0(j+1+(nrp+4)*(i-1)) = u*l1(j,i)+(1-u)*l0(j,i);
        x(d,i+1) = u*x(d,i+1)+(1-u)*x(d,i);
    end
    s2 = s2+toc;
    waitbar(i/ite,P,['\textbf{Remaining time (min): }',...
        num2str((s2/i)*(ite-i)/60,2)])
    if i > ite
        break
    end
    i = i+1;
end
close(P)
%% Input
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '22. Metropolis-Hastings algorithm input'));
save('ite','ite'); save('bii','bii'); save('var','var');
%% Output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '23. Metropolis-Hastings algorithm output'));
save('x','x'); save('l0','l0'); save('l1','l1'); save('u','u');
save('r','r');
```

5.3.5 Projected Newton-Raphson method

```
%% Non-Markovian Projected Newton Raphson method (~1 hour)
clc
clear
close all
tic
%% Estimation input
eps = 5e-5;
sc1 = 1e-2;
sc2 = 2e1-1;
Ni = 2e1;
%% Modified thinning simulation input
path = matlab.desktop.editor.getActiveFilename;
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '1. Modified thinning simulation input'));
ma = load('a'); ma = ma.a; mb = load('b'); mb = mb.b; load('lig');
load('T');
%% Modified thinning simulation output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '2. Modified thinning simulation output'));
load('m'); t = load('t1'); t = t.t1; load('dim'); load('I');
%% Markovian maximum likelihood estimator output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '8. Exact maximum likelihood estimator output'));
load('mle'); load('ver'); load('hor'); load('nrp');
%% Markovian projected Newton-Raphson method input
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '14. Exact projected Newton-Raphson method output'));
load('v');
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2)));
%% Preallocations
[Qm,am,bm] = deal(cell(sc2,Ni));
[Km,Lm] = deal(zeros(1,Ni));
%% Non-Markovian Projected Newton Raphson method
for l = 1:Ni
    x = zeros(nrp+4,1);
    k = 0;
    a = exprnd(ma);
    b = exprnd(mb);
    Q = mle;
    for i = 1:nrp
        x(i) = Q(ver(i),hor(i));
    end
    x(nrp+1:nrp+2) = a;
    x(nrp+3:nrp+4) = b;
    Qm{1,l} = Q; am{1,l} = a; bm{1,l} = b;
    g = fg_loggradientQabfunction(m,t,Q,a,b,v);
    h = fh_loghessianQabfunction(m,t,Q,a,b,v);
    y = h\g;
    x = ((x-y)>eps).*(x-y)+((x-y)<=eps)*eps;
    for j = 1:nrp
        Q(ver(j),hor(j)) = x(j);
    end
    Q(eye(dim)==1) = -sum(Q,2)+Q(eye(dim)==1);
    a = x(nrp+1:nrp+2);
    b = x(nrp+3:nrp+4);
    k = k+1;
    e = fg_loggradientQabfunction(m,t,Q,a,b,v);
    Qm{k+1,l} = Q; am{k+1,l} = a; bm{k+1,l} = b;
    while (max(abs(y)) >= sc1 || max(abs(e-g)) >= sc1) && k <= sc2
```

```
        g = e;
        h = fh_loghessianQabfunction(m,t,Q,a,b,v);
        y = h\g;
        x = ((x-y)>eps).*(x-y)+((x-y)<=eps)*eps;
        for j = 1:nrp
            Q(ver(j),hor(j)) = x(j);
        end
        Q(eye(dim)==1) = -sum(Q,2)+Q(eye(dim)==1);
        a = x(nrp+1:nrp+2);
        b = x(nrp+3:nrp+4);
        k = k+1;
        e = fg_loggradientQabfunction(m,t,Q,a,b,v);
        Qm{k+1,l} = Q; am{k+1,l} = a; bm{k+1,l} = b;
    end
    Lm(l) = fi_loglikelihoodfunction(m,t,Q,a,b,v);
    Km(l) = k;
    disp(l);
end
%% Input
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '16. Non-markovian projected Newton-Raphson method input'));
save('eps','eps'); save('sc1','sc1'); save('sc2','sc2'); save('Ni','Ni');
%% Output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '17. Non-markovian projected Newton-Raphson method output'));
save('Qm','Qm'); save('am','am'); save('bm','bm');
save('Lm','Lm'); save('Km','Km');
toc
```

5.3.6 Discretized simulation

```
%% Non-Markovian discretized simulation (~3 seconds)
clc
clear
close all
tic
%% Simulation input
Q = [-0.2  0.1  0.1;
      0.1 -0.2  0.1;
      0.0  0.0  0.0];
a = 0.1;
b = 1;
T = 10;
dt = 1e-2;
lig = 2;
lve = {'A','B','C'};
Nj = sum(triu(Q)>0,2);
%% Computations
N = round(T/dt);
dim = length(Q);
Q(eye(dim)==1) = -sum(Q,2)+Q(eye(dim)==1);
Qb = eye(dim)+Q*dt;
Qd = triu(Qb)-diag(diag(Qb));
Qu = Qb-Qd;
Qb1 = Qb;
Qb1(2:dim,:) = 0;
Qd1 = Qd;
Qd1(2:dim,:) = 0;
Qu1 = Qu;
Qu1(2:dim,:) = 0;
%% Preallocations
[Phn,Psn] = deal(zeros(dim,dim,N+1));
[P1,P2] = deal(zeros(dim,dim,N+1));
%% Non-Markovian discretized simulation
P1(:,:,1) = eye(dim);
P2(:,:,1) = eye(dim);
P1(:,:,2) = Qb;
P2(:,:,2) = Qb;
Psn(:,:,2) = Qd1*exp(-b*dt);
dum1 = sum(Psn(:,:,2),1);
Phn(:,:,2) = a*dt*[-dum1(1) dum1(1)/2-dum1(2) dum1(1)/2+dum1(2);
                    0       dum1(1)/2-dum1(2) dum1(1)/2+dum1(2);
                    0       0                 0];
for n = 3:N+1
    Psn(:,:,n) = Psn(:,:,n-1)*(Qb*exp(-b*dt)+Qd);
    dum2 = sum(Psn(:,:,n),1);
    dum3 = sum(Qu1*Qu^(n-3)*Qd*exp(-b*dt),1);
    dum4 = sum(Phn(:,:,n-1)*exp(-(n-1)*b*dt),1);
    dum5 = a*dt*[-dum2(1) dum2(1)/2-dum2(2) dum2(1)/2+dum2(2);
                  0       dum2(1)/2-dum2(2) dum2(1)/2+dum2(2);
                  0       0                 0];
    dum6 = a*dt*[-dum3(1) dum3(1)/2-dum3(2) dum3(1)/2+dum3(2);
                  0       dum3(1)/2-dum3(2) dum3(1)/2+dum3(2);
                  0       0                 0];
    dum7 = a*dt*[-dum4(1) dum4(1)/2-dum4(2) dum4(1)/2+dum4(2);
                  0       dum4(1)/2-dum4(2) dum4(1)/2+dum4(2);
                  0       0                 0];
    Phn(:,:,n) = dum5+dum6+dum7+Phn(:,:,n-1)*Qb;
    P1(:,:,n) = Qb^n;
    P2(:,:,n) = Qb^n+sum(Phn(:,:,1:n),3);
```

```
end
%% Input
path = matlab.desktop.editor.getActiveFilename;
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '28. Non-Markovian discretized simulation input'));
save('Q','Q'); save('a','a'); save('b','b'); save('T','T');
save('dt','dt'); save('lig','lig'); save('lve','lve');
%% Output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '29. Non-Markovian discretized simulation output'));
save('P1','P1'); save('P2','P2'); save('dim','dim'); save('Nj','Nj');
toc
```

5.3.7 Heuristic estimator

```matlab
%% Non-Markovian heuristic estimator (~3 seconds)
clc
clear
close all
tic
%% Modified thinning simulation input
path = matlab.desktop.editor.getActiveFilename;
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '1. Modified thinning simulation input'));
load('Q'); load('a'); load('b'); load('lig'); load('dt'); load('T');
load('lve');
%% Modified thinning simulation output
path = matlab.desktop.editor.getActiveFilename;
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '2. Modified thinning simulation output'));
load('dim'); load('Nj');
%% Computations
O = round(T/dt);
t1 = linspace(0,T,O+1);
t2 = t1.';
%% Preallocations
[P1,P2] = deal(zeros(dim-1,dim,dim,O+1));
%% Heuristic estimator
for h = 1:dim-1
    [w,x] = deal(zeros(dim,O+1));
    for i = h:dim-1
        u = -(Q(i,i)+w(i,:)).*exp((Q(i,i)-w(i,:)).*t1);
        v = a((i>lig)+1)*exp(-b((i>lig)+1)*(t2-t1)).*u;
        w(i+1,:) = dt*t1.*(sum(triu(v.'),1)-(diag(v).'+v(:,1).')/2);
        for j = 1:i-1
            x(i,:) = x(i,:)+w(j+1,:).*(Q(j,i)+x(j,:)/Nj(j))./...
                (-Q(j,j)+x(j,:))/Nj(i);
        end
    end
    x(1,:) = zeros(1,O+1);
    x(dim,:) = zeros(1,O+1);
    P1(h,:,:,1) = zeros(dim);
    P1(h,h,h,1) = 1;
    P2(h,:,:,1) = P1(h,:,:,1);
    for i = 1:O
        P1(h,:,:,i+1) = squeeze(P1(h,:,:,i))*...
            (eye(dim)+(Q+(triu(Q>0)-eye(dim).*Nj.').*x(:,i))*dt);
        P2(h,:,:,i+1) = squeeze(P2(h,:,:,i))*(eye(dim)+Q*dt);
    end
end
%% Output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '38. Non-Markovian heuristic estimator output'));
save('P1','P1'); save('P2','P2');
toc
```

5.3.8 Modified Markovian model

```matlab
%% Modified Markovian maximum likelihood estimator (~2 minutes)
clc
clear
close all
tic
%% Modified thinning simulation input
path = matlab.desktop.editor.getActiveFilename;
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '1. Modified thinning simulation input'));
load('lig'); load('T'); load('N'); load('dt');
%% Modified thinning simulation output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '2. Modified thinning simulation output'));
load('m'); load('t1'); load('I'); load('dim');
%% Exact discretized simulation output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '7. Exact maximum likelihood estimator input'));
load('x');
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2)));
%% Preallocations
Mv2 = zeros(1,dim);
mle = cell(dim-1,1);
%% Computations
M = sum(N);
O = round(T/dt);
for i = 1:M
    Mv2(m{i}(1)+1) = Mv2(m{i}(1)+1)+1;
end
Mv1 = cumsum(Mv2);
%% Modified maximum likelihood estimator
parfor i = 1:dim-1
    Nij = zeros(dim);
    Ri = zeros(dim,1);
    for j = Mv1(i)+1:Mv1(i+1)
        for k = 1:dim-1
            Ri(k) = Ri(k)+sum((m{j}==k).*diff(t1{j}));
            for l = 1:dim
                Nij(k,l) = Nij(k,l)+sum((m{j}(1:I(j)-1)==k).*...
                    (m{j}(2:I(j))==l));
            end
        end
    end
    mle{i} = Nij./Ri;
    mle{i}(eye(dim)==1) = -sum(mle{i},2)+mle{i}(eye(dim)==1);
    mle{i}(isnan(mle{i})) = 0;
    mle{i}(dim,:) = 0;
end
%% Preallocations
v = cell(1,12);
C = cell(dim,dim);
dpv = cell(O+1,dim-1);
dpv = cell(O+1,dim);
pme = zeros(dim-1,O+1,dim,dim);
pmi = zeros(dim-1,O+1,dim);
f = cell(1,dim-1);
%% Confidence intervals
for i = 1:dim-1
    m2 = m(Mv1(i)+1:Mv1(i+1));
    t2 = t1(Mv1(i)+1:Mv1(i+1));
```

```
    nrp = sum(mle{i}>0,'all');
    [ver,hor,~,Nj,~] = fb_allowedpairsfunction(m2,dim,1);
    [t2,~,~,~,~,~,~,~,v6] = fc_vectorfunction(m2,t2,ver,hor,Nj,lig);
    v{1} = Mv2(i+1); v{2} = nrp; v{12} = v6;
    v{3} = I(Mv1(i)+1:Mv1(i+1)); v{6} = cell(Mv1(i+1)-Mv1(i));
    h = fd_markovianloghessianfunction(m2,mle{i},v);
    f{i} = -inv(h);
    for j = 1:dim
        ej = zeros(dim,1); ej(j) = 1;
        for k = 1:dim
            ek = zeros(dim,1); ek(k) = 1;
            C{j,k} = [mle{i} ej*ek.'-ej*ej.'; zeros(dim) mle{i}];
        end
    end
    for j = 1:O+1
        t3 = (j-1)*dt;
        for k = 1:dim
            dpv{j,k} = zeros(1,nrp);
            for l = 1:nrp
                dum = expm(C{ver(l),hor(l)}*t3);
                dpv{j,k}(l) = dum(k,2*dim);
            end
        end
        pme(i,j,:,:) = expm(mle{i}*(j-1)*dt);
        for k = 1:dim
            pmi(i,j,k) = norminv(x)*sqrt(dpv{j,k}*f{i}*dpv{j,k}.');
        end
    end
end
%% input
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '34. Modified markovian maximum likelihood estimator input'));
save('dt','dt'); save('x','x');
%% Output
cd(strcat(path(1:strlength(path)-strlength(mfilename)-2),...
    '35. Modified markovian maximum likelihood estimator output'));
save('mle','mle'); save('pme','pme'); save('pmi','pmi');
toc
```

5.3.9 Allowed pairs function

```
function [ver,hor,nrp,Nj,Q] = fb_allowedpairsfunction(m,dim,eps)
    [Q,ver,hor] = deal(zeros(dim));
    nrp = 0;
    M = size(m,1);
    for i = 1:dim
        for j = 1:dim
            if i ~= j
                for k = 1:M
                    if ~isempty(strfind(m{k},[i j]))
                        nrp = nrp+1;
                        Q(i,j) = eps;
                        ver(nrp) = i;
                        hor(nrp) = j;
                        break
                    end
                end
            end
        end
    end
    ver = nonzeros(ver);
    hor = nonzeros(hor);
    Q(eye(dim)==1) = -sum(Q,2)+Q(eye(dim)==1);
    Nj = sum(triu(Q)>0,2).';
end
```

5.3.10 Vectorization function

```
function [t,rv,Nv,v1,v2,v3,v4,v5,v6] = fc_vectorfunction(m,t,ver,hor,Nj,...
    lig)
    M = size(m,1);
    dim = max([ver; hor]);
    nrp = size(ver,1);
    I = zeros(M,1);
    [rv,Nv] = deal(cell(M,1));
    [v1,v2,v3,v4] = deal(cell(M,1));
    [v5,v6] = deal(cell(M,nrp));
    parfor i = 1:M
        I(i) = length(m{i});
        rv{i} = zeros(1,I(i)-1);
        Nv{i} = Nj(m{i}(1:I(i)-1));
        t{i}(I(i)+1) = t{i}(I(i)+1)+(t{i}(I(i))-t{i}(I(i)+1))*...
            (m{i}(I(i))==dim);
        v1{i} = (diff(m{i})>0).*(m{i}(1:I(i)-1)<=lig);
        v2{i} = (diff(m{i})>0).*(m{i}(1:I(i)-1)>lig);
        v3{i} = m{i}(I(i)) == dim && m{i}(I(i)) <= lig;
        v4{i} = m{i}(I(i)) == dim && m{i}(I(i)) > lig;
        for j = 1:nrp
            v5{i,j} = zeros(1,I(i));
            v6{i,j} = zeros(1,I(i)-1);
            for k = 1:I(i)-1
                if length(m{i}) > 1
                    if m{i}(k) == ver(j)
                        v5{i,j}(k) = 1;
                        if m{i}(k+1) == hor(j)
                            v6{i,j}(k) = 1;
                        end
                    end
                end
            end
            v5{i,j}(I(i)) = m{i}(I(i)) == ver(j);
        end
    end
end
```

5.3.11 Logarithmic Markovian gradient function

```
function g = fe_markovianloggradientfunction(m,t,Q,v)
    M = v{1}; nrp = v{2}; I = v{3}; rig = v{4};  Nv = v{5}; Qv = v{6};
    v1 = v{7}; v2 = v{8}; v3 = v{9}; v4 = v{10}; v5 = v{11}; v6 = v{12};
    rsg = rig;
    [m1,m2] = deal(zeros(M,nrp));
    parfor i = 1:M
        if I(i) > 1
            Qv{i} = diag(Q(m{i}(1:I(i)-1),m{i}(1:I(i)-1)+diff(m{i}))).';
            for j = 1:nrp
                m1(i,j) = sum(-v5{i,j}.*diff(t{i}));
                m2(i,j) = sum(v6{i,j}./(Qv{i}));
            end
        else
            for j = 1:nrp
                m1(i,j) = sum(-v5{i,j}.*diff(t{i}));
            end
        end
    end
    g = sum(m1+m2,1).';
end
```

5.3.12 Logarithmic Markovian hessian function

```
function h = fd_markovianloghessianfunction(m,Q,v)
    M = v{1}; nrp = v{2}; I = v{3}; rig1 = v{4}; Nv = v{5}; Qv = v{6};
    v1 = v{7}; v2 = v{8}; v3 = v{9}; v4 = v{10}; v5 = v{11}; v6 = v{12};
    m1 = zeros(M,nrp);
    parfor i = 1:M
        if I(i) > 1
            Qv{i} = diag(Q(m{i}(1:I(i)-1),m{i}(1:I(i)-1)+diff(m{i}))).';
            for j = 1:nrp
                m1(i,j) = sum(-v6{i,j}./Qv{i}.^2);
            end
        end
    end
    h = diag(sum(m1,1));
end
```

5.3.13 Logarithmic Markovian likelihood function

```
function l = ff_markovianloglikelihoodfunction(m,t,Q,v)
    M = v{1}; nrp = v{2}; I = v{3}; rig1 = v{4};  Nv = v{5}; Qv1 = v{6};
    v1 = v{7}; v2 = v{8}; v3 = v{9}; v4 = v{10}; v5 = v{11}; v6 = v{12};
    rsg1 = rig1;
    Qv2 = Qv1;
    [m1,m2,m3] = deal(zeros(M,1));
    parfor i = 1:M
            Qv1{i} = diag(Q(m{i},m{i})).';
            Qv2{i} = diag(Q(m{i}(1:I(i)-1),m{i}(1:I(i)-1)+diff(m{i}))).';
            if I(i) > 1
                m1(i) = sum(Qv1{i}.*diff(t{i}));
                m2(i) = sum(log(Qv2{i}));
            else
                m3(i) = sum(Qv1{i}.*diff(t{i}));
            end
    end
    l = sum(m1+m2+m3);
end
```

5.3.14 Logarithmic non-Markovian gradient function

```
function g = fg_loggradientQabfunction(m,t,Q,a,b,v)
    M = v{1}; nrp = v{2}; I = v{3}; rig1 = v{4};  Nv = v{5}; Qv = v{6};
    v1 = v{7}; v2 = v{8}; v3 = v{9}; v4 = v{10}; v5 = v{11}; v6 = v{12};
    rsg1 = rig1; rsg2 = rsg1; rsg3 = rsg1; rig2 = rig1; rig3 = rig1;
    [m1,m2] = deal(zeros(M,nrp));
    [m3,m4,m5,m6,m7,m8,m9,m10] = deal(zeros(1,M));
    parfor i = 1:M
        if I(i) > 1
            rig3{i}(1) = v1{i}(1)*(t{i}(I(i)+1)-t{i}(2))*...
                exp(-b(1)*(t{i}(I(i)+1)-t{i}(2)));
            rsg3{i}(1) = v2{i}(1)*(t{i}(I(i)+1)-t{i}(2))*...
                exp(-b(2)*(t{i}(I(i)+1)-t{i}(2)));
            for j = 2:I(i)-1
                rig1{i}(j) = exp(-b(1)*(t{i}(j+1)-t{i}(j)))*...
                    (v1{i}(j-1)+rig1{i}(j-1));
                rsg1{i}(j) = exp(-b(2)*(t{i}(j+1)-t{i}(j)))*...
                    (v2{i}(j-1)+rsg1{i}(j-1));
                for k = 1:j-1
                    rig2{i}(j) = rig2{i}(j)+v1{i}(k)*...
                        (t{i}(j+1)-t{i}(k+1))*...
                        exp(-b(1)*(t{i}(j+1)-t{i}(k+1)));
                    rsg2{i}(j) = rsg2{i}(j)+v2{i}(k)*...
                        (t{i}(j+1)-t{i}(k+1))*...
                        exp(-b(2)*(t{i}(j+1)-t{i}(k+1)));
                end
                rig3{i}(j) = v1{i}(j)*(t{i}(I(i)+1)-t{i}(j+1))*...
                    exp(-b(1)*(t{i}(I(i)+1)-t{i}(j+1)));
                rsg3{i}(j) = v2{i}(j)*(t{i}(I(i)+1)-t{i}(j+1))*...
                    exp(-b(2)*(t{i}(I(i)+1)-t{i}(j+1)));
            end
            Qv{i} = diag(Q(m{i}(1:I(i)-1),m{i}(1:I(i)-1)+diff(m{i}))).';
            for j = 1:nrp
                m1(i,j) = sum(-v5{i,j}.*diff(t{i}));
                m2(i,j) = sum(v6{i,j}./(Qv{i}+(v1{i}+v2{i}).*...
                    (a(1).*rig1{i}+a(2).*rsg1{i})./Nv{i}));
            end
                m3(i) = -(sum(v1{i})-v3{i}-...
                    (rig1{i}(I(i)-1)+v1{i}(I(i)-1)-v3{i})*...
                    exp(-b(1)*(t{i}(I(i)+1)-t{i}(I(i)))))/b(1);
                m4(i) = sum(((v1{i}+v2{i}).*rig1{i}./Nv{i})./(Qv{i}+...
                    (a(1)*rig1{i}+a(2).*rsg1{i})./Nv{i}));
                m5(i) = -(sum(v2{i})-v4{i}-...
                    (rsg1{i}(I(i)-1)+v2{i}(I(i)-1)-v4{i})*...
                    exp(-b(2)*(t{i}(I(i)+1)-t{i}(I(i)))))/b(2);
                m6(i) = sum(((v1{i}+v2{i}).*rsg1{i}./Nv{i})./(Qv{i}+...
                    (a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}));
                m7(i) = ((sum(v1{i})-v3{i})*a(1)/b(1)^2)-a(1)/b(1)^2*...
                    (rig1{i}(I(i)-1)+v1{i}(I(i)-1)-v3{i})*...
                    exp(-b(1)*(t{i}(I(i)+1)-t{i}(I(i))))-...
                    (a(1)/b(1)*sum(rig3{i}));
                m8(i) = -sum((a(1)*(v1{i}+v2{i}).*rig2{i}./Nv{i})./...
                    (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}));
                m9(i) = ((sum(v2{i})-v4{i})*a(2)/b(2)^2)-a(2)/b(2)^2*...
                    (rsg1{i}(I(i)-1)+v2{i}(I(i)-1)-v4{i})*...
                    exp(-b(2)*(t{i}(I(i)+1)-t{i}(I(i))))-...
                    (a(2)/b(2)*sum(rsg3{i}));
               m10(i) = -sum((a(2)*(v1{i}+v2{i}).*rsg2{i}./Nv{i})./...
                    (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}));
        else
```

```
            for j = 1:nrp
                m1(i,j) = sum(-v5{i,j}.*diff(t{i}));
            end
        end
    end
    g = [sum(m1+m2,1).'; sum(m3+m4); sum(m5+m6); sum(m7+m8); sum(m9+m10)];
end
```

5.3.15 Logarithmic non-Markovian Hessian function

```
function h = fh_loghessianQabfunction(m,t,Q,a,b,v)
    M = v{1}; nrp = v{2}; I = v{3}; rig1 = v{4}; Nv = v{5}; Qv = v{6};
    v1 = v{7}; v2 = v{8}; v3 = v{9}; v4 = v{10}; v5 = v{11}; v6 = v{12};
    rsg1 = rig1;
    rig2 = rig1; rig3 = rig1; rig4 = rig1; rig5 = rig1;
    rsg2 = rsg1; rsg3 = rsg1; rsg4 = rsg1; rsg5 = rsg1;
    [m1,m2,m3,m4,m5] = deal(zeros(M,nrp));
    [m6,m7,m8,m9,m10,m11,m12,m13,m14,m15] = deal(zeros(1,M));
    parfor i = 1:M
        if I(i) > 1
            rig3{i}(1) = v1{i}(1)*(t{i}(I(i)+1)-t{i}(2))*...
                exp(-b(1)*(t{i}(I(i)+1)-t{i}(2)));
            rsg3{i}(1) = v2{i}(1)*(t{i}(I(i)+1)-t{i}(2))*...
                exp(-b(2)*(t{i}(I(i)+1)-t{i}(2)));
            rig5{i}(1) = v1{i}(1)*(t{i}(I(i)+1)-t{i}(2))^2*...
                exp(-b(1)*(t{i}(I(i)+1)-t{i}(2)));
            rsg5{i}(1) = v2{i}(1)*(t{i}(I(i)+1)-t{i}(2))^2*...
                exp(-b(2)*(t{i}(I(i)+1)-t{i}(2)));
            for j = 2:I(i)-1
                rig1{i}(j) = exp(-b(1)*(t{i}(j+1)-t{i}(j)))*...
                    (v1{i}(j-1)+rig1{i}(j-1));
                rsg1{i}(j) = exp(-b(2)*(t{i}(j+1)-t{i}(j)))*...
                    (v2{i}(j-1)+rsg1{i}(j-1));
                for k = 1:j-1
                    rig2{i}(j) = rig2{i}(j)+v1{i}(k)*...
                        (t{i}(j+1)-t{i}(k+1))*exp(-b(1)*...
                        (t{i}(j+1)-t{i}(k+1)));
                    rsg2{i}(j) = rsg2{i}(j)+v2{i}(k)*...
                        (t{i}(j+1)-t{i}(k+1))*exp(-b(2)*...
                        (t{i}(j+1)-t{i}(k+1)));
                    rig4{i}(j) = rig4{i}(j)+v1{i}(k)*...
                        (t{i}(j+1)-t{i}(k+1))^2*exp(-b(1)*...
                        (t{i}(j+1)-t{i}(k+1)));
                    rsg4{i}(j) = rsg4{i}(j)+v2{i}(k)*...
                    (t{i}(j+1)-t{i}(k+1))^2*exp(-b(2)*...
                    (t{i}(j+1)-t{i}(k+1)));
                end
                rig3{i}(j) = v1{i}(j)*(t{i}(I(i)+1)-t{i}(j+1))*...
                    exp(-b(1)*(t{i}(I(i)+1)-t{i}(j+1)));
                rsg3{i}(j) = v2{i}(j)*(t{i}(I(i)+1)-t{i}(j+1))*...
                    exp(-b(2)*(t{i}(I(i)+1)-t{i}(j+1)));
                rig5{i}(j) = v1{i}(j)*(t{i}(I(i)+1)-t{i}(j+1))^2*...
                    exp(-b(1)*(t{i}(I(i)+1)-t{i}(j+1)));
                rsg5{i}(j) = v2{i}(j)*(t{i}(I(i)+1)-t{i}(j+1))^2*...
                    exp(-b(2)*(t{i}(I(i)+1)-t{i}(j+1)));
            end
            Qv{i} = diag(Q(m{i}(1:I(i)-1),m{i}(1:I(i)-1)+diff(m{i}))).';
            for j = 1:nrp
                m1(i,j) = sum(-v6{i,j}./(Qv{i}+(v1{i}+v2{i}).*...
                    (a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2);
                m2(i,j) = sum(-v6{i,j}.*(v1{i}+v2{i}).*rig1{i}./Nv{i}./...
                    (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2);
                m3(i,j) = sum(-v6{i,j}.*(v1{i}+v2{i}).*rsg1{i}./Nv{i}./...
                    (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2);
                m4(i,j) = sum(a(1)*v6{i,j}.*(v1{i}+v2{i}).*rig2{i}./...
                    Nv{i}./(Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2);
                m5(i,j) = sum(a(2)*v6{i,j}.*(v1{i}+v2{i}).*rsg2{i}./...
                    Nv{i}./(Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2);
            end
```

```
            m6(i) = sum(-(v1{i}+v2{i}).*(rig1{i}./Nv{i}).^2./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2);
            m7(i) = sum(-(v1{i}+v2{i}).*rig1{i}.*rsg1{i}./Nv{i}.^2./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2);
            m8(i) = sum(-(v1{i}+v2{i}).*rig2{i}./Nv{i}./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i})+...
                a(1)*(v1{i}+v2{i}).*rig1{i}.*rig2{i}./Nv{i}.^2./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2)-...
                sum(rig3{i})/b(1)+...
                (sum(v1{i})-v3{i})/b(1)^2-...
                (rig1{i}(I(i)-1)+v1{i}(I(i)-1)-v3{i})*...
                exp(-b(1)*(t{i}(I(i)+1)-t{i}(I(i))))/b(1)^2;
            m9(i) = sum(a(2)*(v1{i}+v2{i}).*rig1{i}.*rsg2{i}./Nv{i}.^2./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2);
            m10(i) = sum(-(v1{i}+v2{i}).*(rsg1{i}./Nv{i}).^2./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2);
            m11(i) = sum(a(1)*(v1{i}+v2{i}).*rsg1{i}.*rig2{i}./Nv{i}.^2./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2);
            m12(i) = sum(-(v1{i}+v2{i}).*rsg2{i}./Nv{i}./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i})+...
                a(2)*(v1{i}+v2{i}).*rsg1{i}.*rsg2{i}./Nv{i}.^2./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2)-...
                sum(rsg3{i})/b(2)+...
                (sum(v2{i})-v4{i})/b(2)^2-...
                (rsg1{i}(I(i)-1)+v2{i}(I(i)-1)-v4{i})*...
                exp(-b(2)*(t{i}(I(i)+1)-t{i}(I(i))))/b(2)^2;
            m13(i) = sum(a(1)*(v1{i}+v2{i}).*rig4{i}./Nv{i}./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}))-...
                sum((v1{i}+v2{i}).*(a(1)*rig2{i}./Nv{i}).^2./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2)-...
                (sum(v1{i})-v3{i})*2*a(1)/b(1)^3+...
                2*a(1)/b(1)^3*sum((rig1{i}(I(i)-1)+v1{i}(I(i)-1)-v3{i})*...
                exp(-b(1)*(t{i}(I(i)+1)-t{i}(I(i)))))+...
                2*a(1)/b(1)^2*sum(rig3{i})+...
                a(1)/b(1)*sum(rig5{i});
            m14(i) = sum(-a(1)*a(2).*rig2{i}.*rsg2{i}.*(v1{i}+v2{i})./...
                Nv{i}.^2./(Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2);
            m15(i) = sum(a(2)*(v1{i}+v2{i}).*rsg4{i}./Nv{i}./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}))-...
                sum((v1{i}+v2{i}).*(a(2)*rsg2{i}./Nv{i}).^2./...
                (Qv{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./Nv{i}).^2)-...
                (sum(v2{i})-v4{i})*2*a(2)/b(2)^3+...
                2*a(2)/b(2)^3*sum((rsg1{i}(I(i)-1)+v2{i}(I(i)-1)-v4{i})*...
                exp(-b(2)*(t{i}(I(i)+1)-t{i}(I(i)))))+...
                2*a(2)/b(2)^2*sum(rsg3{i})+...
                a(2)/b(2)*sum(rsg5{i});
        end
    end
    h = ...
    [diag(sum(m1,1)) sum(m2,1).' sum(m3,1).' sum(m4,1).' sum(m5,1).';...
        sum(m2,1) sum(m6) sum(m7)  sum(m8)  sum(m9);...
        sum(m3,1) sum(m7) sum(m10) sum(m11) sum(m12);...
        sum(m4,1) sum(m8) sum(m11) sum(m13) sum(m14);...
        sum(m5,1) sum(m9) sum(m12) sum(m14) sum(m15)];
end
```

5.3.16 Logarithmic non-Markovian likelihood function

```
function l = fi_loglikelihoodfunction(m,t,Q,a,b,v)
    M = v{1}; nrp = v{2}; I = v{3}; rig1 = v{4};  Nv = v{5}; Qv1 = v{6};
    v1 = v{7}; v2 = v{8}; v3 = v{9}; v4 = v{10}; v5 = v{11}; v6 = v{12};
    rsg1 = rig1;
    Qv2 = Qv1;
    [m1,m2,m3,m4,m5] = deal(zeros(M,1));
    parfor i = 1:M
            for j = 2:I(i)-1
                rig1{i}(j) = exp(-b(1)*(t{i}(j+1)-t{i}(j)))*...
                    (v1{i}(j-1)+rig1{i}(j-1));
                rsg1{i}(j) = exp(-b(2)*(t{i}(j+1)-t{i}(j)))*...
                    (v2{i}(j-1)+rsg1{i}(j-1));
            end
            Qv1{i} = diag(Q(m{i},m{i})).';
            Qv2{i} = diag(Q(m{i}(1:I(i)-1),m{i}(1:I(i)-1)+diff(m{i}))).';
            if I(i) > 1
                m1(i) = sum(Qv1{i}.*diff(t{i}));
                m2(i) = -a(1)/b(1)*(sum(v1{i})-v3{i}-...
                    (rig1{i}(I(i)-1)+v1{i}(I(i)-1)-v3{i})*...
                    exp(-b(1)*(t{i}(I(i)+1)-t{i}(I(i)))));
                m3(i) = -a(2)/b(2)*(sum(v2{i})-v4{i}-...
                    (rsg1{i}(I(i)-1)+v2{i}(I(i)-1)-v4{i})*...
                    exp(-b(2)*(t{i}(I(i)+1)-t{i}(I(i)))));
                m4(i) = sum(log(Qv2{i}+(a(1)*rig1{i}+a(2)*rsg1{i})./...
                    Nv{i}.*(v1{i}+v2{i})));
            else
                m5(i) = sum(Qv1{i}.*diff(t{i}));
            end
    end
    l = sum(m1+m2+m3+m4+m5);
end
```

Bibliography

[1] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables.* US Government printing office, 1964.

[2] Arthur Albert. "Estimating the infinitesimal generator of a continuous time, finite state Markov process". In: *The Annals of Mathematical Statistics* (1962), pp. 727–753.

[3] David E Allen and Robert J Powell. *Credit risk measurement methodologies.* 2011.

[4] Edward I Altman and Duen-Li Kao. "The implications of corporate bond ratings drift". In: *Financial Analysts Journal* 48.3 (1992), pp. 64–75.

[5] Adi Ben-Israel. "A Newton-Raphson method for the solution of systems of equations". In: *Journal of Mathematical analysis and applications* 15.2 (1966), pp. 243–252.

[6] Tomasz R Bielelcki, Stéphane Crépey, and Alexander Herbertsson. "Markov chain models of portfolio credit risk". In: *The Oxford Handbook of Credit Derivatives* (2011).

[7] Mogens Bladt and Michael Sørensen. "Efficient estimation of transition rates between credit ratings from observations at discrete time points". In: *Quantitative Finance* 9.2 (2009), pp. 147–160.

[8] Mogens Bladt and Michael Sørensen. "Statistical inference for discretely observed Markov jump processes". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.3 (2005), pp. 395–410.

[9] Olivier Cappé, Eric Moulines, and Tobias Rydén. "Inference in hidden markov models". In: *Proceedings of EUSFLAT conference.* 2009, pp. 14–16.

[10] Mark Carey and Mark Hrycay. "Parameterizing credit risk models with rating data". In: *Journal of banking & finance* 25.1 (2001), pp. 197–270.

[11] Marta Casanellas, Jesús Fernández-Sánchez, and Jordi Roca-Lacostena. *The embedding problem for Markov matrices.* 2021.

[12] Joe Chang. *Book manuscript, course: Stochastic processes at Yale university.* 2021.

[13] Yuanda Chen. "Thinning algorithms for simulating point processes". In: *Florida State University, Tallahassee, FL* (2016).

[14] Jens HE Christensen, Ernst Hansen, and David Lando. "Confidence sets for continuous-time rating transition probabilities". In: *Journal of Banking & Finance* 28.11 (2004), pp. 2575–2602.

[15] Fabien Couderc. *Credit risk and ratings: Understanding dynamics and relationships with macroeconomics.* 2008.

[16] Guglielmo D'Amico, Jacques Janssen, and Raimondo Manca. "Downward migration credit risk problem: a non-homogeneous backward semi-Markov reliability approach". In: *Journal of the Operational Research Society* 67.3 (2016), pp. 393–401.

[17] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume I: elementary theory and methods.* 2nd ed. Springer New York, 2003.

[18] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: General Theory and Structure.* 2nd ed. Springer New York, 2007.

[19] Angelos Dassios and Hongbiao Zhao. "Exact simulation of Hawkes process with exponentially decaying intensity". In: *Electronic Communications in Probability* 18 (2013), pp. 1–13.

[20] Frederik M Dekking et al. *A Modern Introduction to Probability and Statistics: Understanding why and how.* Springer Science & Business Media, 2005.

[21] Goncalo Dos Reis, Marius Pfeuffer, and Greig Smith. "Capturing model risk and rating momentum in the estimation of probabilities of default and credit rating migrations". In: *Quantitative Finance* 20.7 (2020), pp. 1069–1083.

[22] Goncalo Dos Reis and Greig Smith. "Robust and consistent estimation of generators in credit risk". In: *Quantitative Finance* 18.6 (2018), pp. 983–1001.

[23] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap.* CRC press, 1994.

[24] Eugene Feinberg, Manasa Mandava, and Albert N Shiryaev. "Kolmogorov's equations for jump Markov processes with unbounded jump rates". In: *Annals of Operations Research* (2017), pp. 1–18.

[25] Halina Frydman and Til Schuermann. "Credit rating dynamics and Markov mixture models". In: *Journal of Banking & Finance* 32.6 (2008), pp. 1062–1075.

[26] Paul A Gagniuc. *Markov chains: from theory to implementation and experimentation.* John Wiley & Sons, 2017.

[27] Irving J Good. "Some statistical applications of Poisson's work". In: *Statistical science* (1986), pp. 157–170.

[28] Christian Gourieroux and Alain Monfort. *Statistics and econometric models, volume I.* Cambridge University Press, 1995.

[29] Roger A Horn and Charles R Johnson. *Matrix analysis.* Cambridge university press, 2012.

[30] Yasunari Inamura et al. "Estimating continuous time transition matrices from discretely observed data". In: *Bank of Japan* (2006), pp. 06–07.

[31] Robert B Israel, Jeffrey S Rosenthal, and Jason Z Wei. "Finding generators for Markov chains via empirical transition matrices, with applications to credit ratings". In: *Mathematical finance* 11.2 (2001), pp. 245–265.

[32] Edwin T Jaynes. *Probability theory: The logic of science.* Cambridge university press, 2003.

[33] John D Kalbfleisch and Jerald F Lawless. "The analysis of panel data under a Markov assumption". In: *Journal of the American statistical association* 80.392 (1985), pp. 863–871.

[34] David G Kleinbaum and Mitchel Klein. *Survival analysis.* 2nd ed. Springer, 2012.

[35] Donald E Knuth. "Big omicron and big omega and big theta". In: *ACM Sigact News* 8.2 (1976), pp. 18–24.

[36] Małgorzata W Korolkiewicz. "A dependent hidden Markov model of credit quality". In: *International Journal of Stochastic Analysis* 2012 (2012).

[37] David Lando and Torben M Skødeberg. "Analyzing rating transitions and rating drift with continuous observations". In: *Journal of banking & finance* 26.2-3 (2002), pp. 423–444.

[38] Erich L Lehmann and George Casella. *Theory of point estimation.* 2nd ed. Springer Science & Business Media, 2006.

[39] Emmanuel Lesaffre and Andrew B Lawson. *Bayesian biostatistics.* 1st ed. John Wiley & Sons, 2012.

[40] Gunter Löffler. "Avoiding the rating bounce: Why rating agencies are slow to react to new information". In: *Journal of Economic Behavior & Organization* 56.3 (2005), pp. 365–381.

[41] Alexander J McNeil, Rüdiger Frey, and Paul Embrechts. *Quantitative risk management: concepts, techniques and tools-revised edition.* Princeton university press, 2015.

[42] Elizabeth Meckes. *The Eigenvalues of Random Matrices.* 2021.

[43] Pamela Nickell, William Perraudin, and Simone Varotto. "Stability of rating transitions". In: *Journal of Banking & Finance* 24.1-2 (2000), pp. 203–227.

[44] James R Norris. *Markov chains.* 2nd ed. Cambridge university press, 1998.

[45] David Oakes. "Direct calculation of the information matrix via the EM". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.2 (1999), pp. 479–482.

[46] Yosihiko Ogata. "On Lewis' simulation method for point processes". In: *IEEE transactions on information theory* 27.1 (1981), pp. 23–31.

[47] Andrea Pascucci. *PDE and martingale methods in option pricing.* Springer Science & Business Media, 2011.

[48] Marius Pfeuffer. "ctmcd: An R Package for Estimating the Parameters of a Continuous-Time Markov Chain from Discrete-Time Data." In: *R Journal* 9.2 (2017).

[49] Bo Ranneby. "On necessary and sufficient conditions for consistency of MLE's in Markov chain models". In: *Scandinavian Journal of Statistics* (1978), pp. 99–105.

[50] Christian P Robert. "Simulation of truncated normal variables". In: *Statistics and computing* 5.2 (1995), pp. 121–125.

[51] Christian P Robert and George Casella. *Monte Carlo statistical methods.* 2nd ed. Springer, 2004.

[52] Sheldon M Ross. *Introduction to probability models.* 10th ed. Elsevier, 2010.

[53] Sheldon M Ross. *Simulation.* 5th ed. Academic press, 2013.

[54] Walter Rudin et al. *Principles of mathematical analysis.* 3rd ed. McGraw-hill New York, 1964.

[55] Mohammed A Salman and V C Borkar. "Exponential Matrix and Their Properties". In: *International Journal of Scientific and Innovative Mathematical Research (IJSIMR) Volume* 4 (2016), pp. 53–63.

[56] Tomáš Vaněk, David Hampel, et al. "The probability of default under ifrs 9: Multi-period estimation and macroeconomic forecast". In: *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis* 65.2 (2017), pp. 759–776.

[57] Guanyang Wang. *Exact convergence analysis of the independent Metropolis-Hastings algorithms*. 2021.

[58] Tetsuro Yamamoto. "Historical developments in convergence analysis for Newton's and Newton-like methods". In: *Journal of Computational and Applied Mathematics* 124.1-2 (2000), pp. 1–23.