UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Markov Processes in Disease Modelling - Estimation and Implementation

by

Christiaan Antonie Marais

Submitted in partial fulfilment for the degree

Magister Scientiae

In the Department of Statistics

In the Faculty of Natural & Agricultural Sciences

University of Pretoria

Pretoria

June 2010

Declaration

I, Christiaan Antonie Marais, declare that the dissertation, which I hereby submit for the degree Magister Scientiae at the University of Pretoria, is my own work and have not previously been submitted by me for a degree at this or another tertiary institution.

SIGNITURE:____________________

DATE:__________________________

Acknowledgements

Jesus Christ, the creator of the universe and my tower of strength. “The Lord is my strength and my shield; my heart trusted in Him, and I am helped. Therefore my heart greatly rejoice; and with song will I praise Him.” Psalm 28:7

Inger Fabris-Rotelli and Prof Sarma Yadavalli : For the continued input and comments during the search for and understanding of literature. Thanks for making time in your busy schedule and for the motivation throughout my work.

Prof Philippe Beutels (Centre for Health Economics Research and Modeling Infectious Diseases (CHERMID) at the University of Antwerp): For providing me access to a computer and printing facilities.

Prof Niel Hens (from CHERMID): For the input and suggestions during our discussions regarding my dissertation.

My family and friends: For your continued support and prayers during my work and especially for the understanding in the last weeks when I had to prepare the final version of my dissertation.

The University of Pretoria: For providing an excellent platform for studies and research throughout my tertiary education. A special word of thanks for providing me with a bursary for my studies and for allowing an extension on the initial term of the bursary.

Researchers past and present: I am indebted to the efforts that has been put into understanding diseases and the estimation of the effects thereof on society. I hope the literature will continue to grow as we seek to extend on the current methods in statistical analysis.

Summary

There exists a need to estimate the potential financial, epidemiological and societal impact that diseases, and the treatment thereof, can have on society. Markov processes are often used to model diseases to estimate these quantities of interest and have an advantage over standard survival analysis techniques in that multiple events can be studied simultaneously. The theory of Markov processes is well established for processes for which the process parameters are known but not as much of the literature has focussed on the estimation of these transition parameters.

This dissertation investigates and implements maximum likelihood estimators for Markov processes based on longitudinal data. The methods are described based on processes that are observed such that all transitions are recorded exactly, processes of which the state of the process is recorded at equidistant time points, at irregular time points and processes for which each process is observed at a possibly different irregular time point. Methods for handling right censoring and estimating the effect of covariates on parameters are described.

The estimation methods are implemented by simulating Markov processes and estimating the parameters based on the simulated data so that the accuracy of the estimators can be investigated. We show that the estimators can provide accurate estimates of state prevalence if the process is stationary, even with relatively small sample sizes. Furthermore, we indicate that the estimators lack good accuracy in estimating the effect of covariates on parameters unless state transitions are recorded exactly. The methods are discussed with reference to the *msm* package for R which is freely available and a popular tool for estimating and implementing Markov processes in disease modelling.

Methods are mentioned for the treatment of aggregate data, diseases where the state of patients are not known with complete certainty at every observation and diseases where patient interaction plays a role.

Table of contents

Chapter 1: Introduction

1.1 Problem statement

There exists a need to estimate the potential financial, epidemiological and societal impact that diseases, and the treatment thereof, can have on society. For example, the Department of Health in South Africa has a limited amount of resources that is being made available to them by the treasury. This budget has to be spent as fairly and effectively as possible and it would therefore be beneficial for the decision makers to have an idea of the potential impact that certain medical interventions may have. As another example, when a healthcare funder like the Department of Health or a medical insurance fund has to decide if they will reimburse a new treatment, it may be of interest to them to estimate the long term effects of such an investment on the morbidity and mortality of the patients receiving the new treatment. The long term financial impact may also be of interest when deciding if the new treatment should be reimbursed.

When comparing the long term effects of a new treatment option with the current treatment of a disease one may want to consider a sequence of disease and / or treatment related events. Cancer patients for example can undergo surgery to have cancer cells removed and thereafter be in a “disease free” state. Then the cancer cells can reappear at the same site (local recurrence) or the cancer cells can spread to another part of the body (metastasis) or the patient can die[87]. We know the patient will die eventually from cancer or other causes of death but one may want like to investigate the time until death for the new and current treatment options, the difference in time until death after a local recurrence and after metastasis, and the possible avoidance or prolonging of a local recurrence or metastasis with the proposed new treatment option.

Longitudinal data consists of repeated measurements of the state of a patient and the time between observations and can therefore be used to answer the research questions posed in the preceding paragraph. Longitudinal data is available from clinical trials since these trials are often done over a period of time with observations being made on patients at regular (or sometimes irregular) time intervals until the patient dies, is removed from the trial or the study ends[5,56,47]. An extract of two patients from a study by Klotz and Sharpless[57] is shown in Table 1 to give an example of longitudinal data. The study by Klotz and Sharpless will be discussed in more detail in Section 2.4 of the dissertation.

Patient ID	**Time (Year)**	**Status**
001	0	Disease free
001	1	Disease free
001	2	Mild disease
001	3	Mild disease
001	4	Mild disease
001	5	Moderate / severe disease
001	6	Death
002	0	Disease free
002	1	Disease free
002	2	Moderate / severe disease
002	3	Death

Table 1: An example of longitudinal data

Standard survival analysis techniques like Kaplan Meier survival curves[55] and Cox regression models[19] can be used when one wishes to study the time until a specified event occurs. This however fails to describe a sequence of events and does therefore not utilise all the information that longitudinal data can provide if multiple events are studied.

Markov processes posses a potential solution to the modelling of a sequence of events with the use of longitudinal data[104] and have been used on several occasions in the literature especially in the field of pharmacoeconomics [85,22 ,113,46]. Markov processes are favourable to the modelling of diseases when a disease can be grouped into a set of exhaustive and mutually exclusive health states, thereby forming a multi-state model.

The theory of Markov processes is well established for processes for which the process parameters are known but not as much of the literature has focussed on the estimation of these transition parameters[10]. Furthermore, various assumptions are often made in terms of the structure of the processes that are used in the construction of Markov processes used for disease modelling. These assumptions include constant risks of events over time and that all subjects represented by the model are homogenous in terms of their risks of future events. These assumptions are made to simplify the implementation of the model, but are sometimes not robust since transition parameters may change over time based on the occurrence of an event. One example of this is the increased risk of death in the first 30 days after a myocardial infarction, but with the risk decreasing thereafter[2] which is a violation of the assumption of constant risks of events over time. The Framingham heart study indicated that, amongst others, smoking, high blood pressure and a history diabetes increases ones risk of having a stroke[23]. This indicates that it will be inappropriate to assume a group of patients all have the same risk of a stroke if the group contains smokers and non-smokers, patients with significantly varying blood pressure levels and some with diabetes and some without diabetes.

Another problem in disease modelling is that the data used to estimate the process parameters often contains censoring. Censoring occurs for example when the time until an event is not observed but known to be greater than some non-zero value. The state of the process can also be censored depending on how frequent the process is observed. Ideally one would like to observe state transitions at the exact time that it occurs, but this is not always practically possible. The state of the process is often observed at some predefined time points and such data is called panel data.

This dissertation discusses methods for implementing Markov processes based on the frequency of observations, with the inclusion of estimating the effects of covariates on the transition parameters. Methods for dealing with transition parameters that change over time are also mentioned.

Markov processes are not suitable for modelling all disease types and answering all disease related questions due to the complexity that is involved in the modelling of some diseases. It is therefore important to highlight situations when researchers should consider alternative methods. This will be discussed in the concluding chapter of the dissertation.

There is a vast amount of literature available on the estimation of the processes and it will not be feasible to discuss all these methods in detail in this dissertation. This dissertation will focus on the first article written to estimate Markov processes with a discrete time variable and the first article written for estimating the parameters of a continuous time process. Then the literature used to construct the *msm* package[49] of R[91] will be discussed since this package is often used in practise when estimating Markov process parameters. The methods will be implemented by simulating processes and estimating their parameters. The methods will then be discussed and compared in terms of accuracy, complexity, and possible shortcomings

The dissertation will conclude by mentioning recent advances in the literature not included in the *msm* package for R that can also be used for estimating Markov process parameters. Other scenarios whereby Markov processes should be avoided for disease modelling will also be discussed.

All computer procedures were written in R[91] and are shown in Appendix 2. These programs were executed on a computer with a Intel® Geon® 3.33 GHz CPU and 24 GB RAM with a 64 bit Windows 7[77] operating system. The computer specifications are provided since comments are made on the run time for some procedures.

1.2 Dissertation outline

This chapter will continue with a literature review which will review the current literature that is available on the topic of Markov processes and the estimation of Markov processes.

Chapter 2 will provide background information for Markov processes which will consist of measure theory, the Markov property, Markov chains and Markov jump processes and finally a brief overview of some survival analysis concepts and a discussion thereof. The terms Markov chains and Markov jump processes will be used in the remainder of Chapter 1 but will only be formally introduced in Chapter 2. Markov chains refer to Markov processes in discrete time and Markov jump processes refer to Markov processes in continuous time, both with discrete state spaces. The term "Markov process" will be used as a collective term for Markov chains and Markov jump processes.

Chapter 3 will review maximum likelihood (ML) estimators for Markov chains and Markov jump processes. The chapter will start with Markov chains since the theory involved is thought to be easily understandable and since this is the type of Markov process for which ML estimators of Markov processes was initiated. Then a Markov jump processes of which all transitions are observed exactly will be discussed. This discussion involves a lot of seemingly complicated measure theory and therefore an alternative derivation of these results for Markov jump processes will be provided which is thought to follow a more comprehensible approach. The assumption of observing all transitions exactly will be relaxed and methods for handling this type of censored data of will be discussed. Notes will be made on how covariates can be included in the methods discussed and how censored time should be handled. Methods for testing the goodness-of-fit models of Markov processes will also be described. The methods discussed in Chapter 3 will focus on the *msm* package for R since, as we show in the literature review, this package is often used in multi-state models.

Chapter 4 will implement the estimation techniques discussed in Chapter 3 by simulating Markov processes with different observation schemes and estimating the parameters thereof based on the simulated data. We will start with the observations that are observed exactly and discuss how covariate effects can be implemented. Then we consider processes that are observed at equidistant times, processes that are observed at the same irregular time points and finally the case where all processes are observed at possibly different irregular time points. We conclude with an example where all processes are observed at possibly different irregular time points with the inclusion of covariate effects. Goodness-of-fit tests will be implemented and discussed throughout to indicate their ability to assess model fit.

Chapter 5 will contain a discussion of the techniques described in the preceding chapters. The discussion will include suggestions on when which types of the discussed estimation methods will be the most suitable in terms of the observation scheme of the data. The discussion will end with an overview of other disease modelling techniques that are available which fall outside the scope of this dissertation and situations when these methods should be considered as opposed to Markov processes.

The flow of the dissertation is illustrated in Figure 1. All chapters will commence with a brief introduction to the contents of the chapter and will be concluded with a summary.

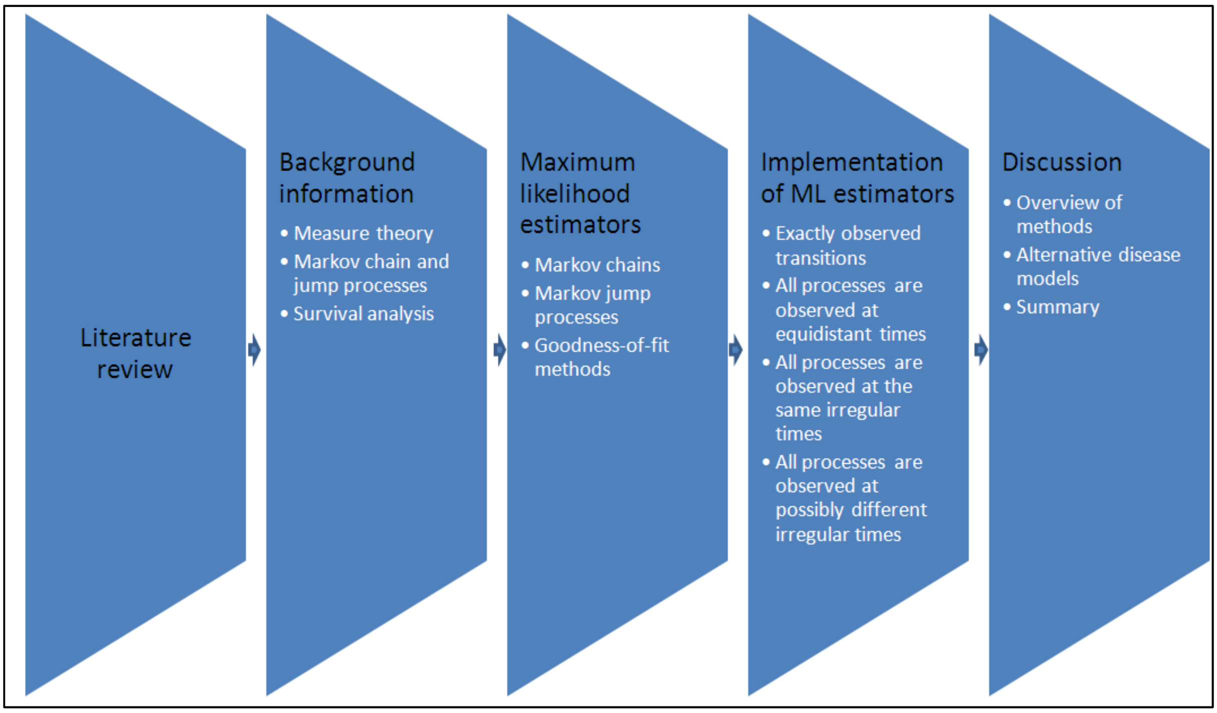
Literature review
Background information
Measure theory
Markov chain and jump processes
Survival analysis
Maximum likelihood estimators
Markov chains
Markov jump processes
Goodness-of-fit methods
Implementation of ML estimators
Exactly observed transitions
All processes are observed at equidistant times
All processes are observed at the same irregular times
All processes are observed at possibly different irregular times
Discussion
Overview of methods
Alternative disease models
Summary

Figure 1: Dissertation outline

1.3 Literature review

The literature review consists of two sections. The first section looks specifically at Masters degree and Doctorate degree dissertations and theses (for the remainder of this dissertation the term thesis will be used to refer to both theses and dissertations unless otherwise specified) that were done at South African universities on the topic of the use of Markov processes in disease modelling. This was done to determine the extent to which postgraduate research has been done in the area of Markov processes in disease modelling at South African Universities.

The aim of the second section of the literature review is to identify all influential papers and books that have contributed to the estimation of the parameters of Markov processes in South Africa and internationally. Due to the topic of this dissertation, preference was given to techniques that are based on clinical data or data that are presented in a similar fashion as clinical data.

The literature review did not include work that focussed on hidden Markov models as this falls outside the scope of this dissertation but some comments will be made about the use of hidden Markov models in disease models in Chapter 5.

The outcome of the literature review for the two sections will be described separately and will be followed by a summary of the two sections.

1.3.1 South African literature on Markov processes for disease modelling

Previous theses at South African universities were searched using the UCTD database which is provided by the University of Cape Town. This database contains all theses from South African universities. The UCTD database was searched using the keywords ‘Markov’ and ‘Disease’ and this yielded no results. The database search was broadened by using only the term ‘Markov’. This exercise yielded 47 results of which 12 were repetitive. None of these theses were focussed on disease modelling. The results from the UCTD database were applications of Markov modelling in speech recognition[32,107,89,92,110,100,111,106 ,63,88,11,101,83,108], optical recognition[79,17,98,114,72,109,59,80], mixed-order hidden Markov models[96,28,52] and others[7,66,105,62,78,76,71,61,112,115] but none of these theses were associated with disease modelling or the estimation of the transition probabilities of standard Markov models.

Next, the UCTD database was searched using the terms ‘Disease modelling’ and this resulted in three theses. One of these theses modelled the co-infection dynamics of HIV-1 and Tuberculosis[27] and the other modelled the relationships between clinical markers of HIV[42]. These two theses did not use Markovian methods. The third thesis, entitled *Stochastic analysis of AIDS epidemiology* by Labeodan MMO[58], is a PhD thesis that contained, amongst others, methods for estimating transition rates between two states of a stochastic model, but it was not focused on the estimation of Markov process parameters based on longitudinal data as intended in this dissertation.

The UCTD database was further searched using the term ‘disease simulation’ which yielded the dissertation by Du Toit[27] on the co-infection of HIV-1 and TB.

With the goal of identifying South African dissertations on the estimation of transition probabilities in Markov processes the UCTD database was searched with the term ‘transition matrix’ and this yielded no results. A search for the term ‘rate matrix' yielded one thesis[82] which did not have any relevance to the Markovian setting.

1.3.2 Literature on the estimation of Markov process probabilities

A literature review was conducted by searching for the terms ‘estimation’ AND ‘transition probabilities’ OR ‘transition rates’ AND ‘Markov process’ on the following databases:

- Statistical theory & methods
- Zentralbalt – MATH
- MathSciNet
- Scopus
- Current index to Statistics
- JStor
- ISI web of knowledge

Results from these searches were filtered to obtain only papers with primary focus on the estimation of the transition probabilities of a Markov process. All papers dealing with hidden Markov processes were disregarded. This resulted in 12 articles. Due to the relatively low number of articles obtained, it was decided to generalise the search by searching for the terms ‘Markov process’ AND ‘estimation’ AND ‘matrix’. This was done on the following databases which are thought to be the most relevant databases for the field of disease modelling:

- Scopus
- Bulletin of Mathematical Biology
- Mathematical Biosciences
- ISI web of knowledge

The last mentioned literature search was limited to articles from the year 2000 onwards to ensure that the most recent articles are obtained. The “classic” articles dating back prior to 2000 for the estimation procedures were identified from the introductory notes of these more recent articles.

1.3.2.1 Outcome of literature search on estimation of transition probabilities

The theory of stochastic processes is summarised in various textbooks and articles. Four of the most referenced textbooks are *The Theory of Stochastic Processes* by Cox et al.[20], *An Introduction to Probability Theory and Its Applications* by Feller[33], *Introduction to Stochastic Processes* by Cinlar[16] and *Stochastic Processes* by Doob[26].

Many articles were found which described the implementation of Markov models in disease modelling. These include the use of Markov processes in modelling hepatitis C[102], cancer[31,56 ,29,15], diabetes[21,6], diabetic retinopathy[41,69] and trachoma infection[39] (both are eye diseases), malaria[90], Human Immunodeficiency Virus (HIV)[35,68,64,95,40,38,74], and cardiovascular diseases[51,97,57].

Markov processes are also used in, amongst others, the social sciences[53,103,104], geology[84,25], marketing[30,24], process control[13], veterinary sciences[37], weather prediction[94] and linguistics[34].

The first and one of the most referenced articles for the estimation of the transition probabilities in Markov chains is the article by Anderson and Goodman which was published in 1957[8]. In this article the likelihood function for a Markov chain is derived and maximised to find the ML estimators for the entries of a transition probability matrix (TPM). The asymptotic behaviour of the estimators is also discussed. Furthermore, methods for testing whether transition probabilities are equal to a specific hypothesised value are discussed.

In 1962 the work of discrete time Markov chain ML estimator of Anderson and Goodman[8] was extended to a continuous time Markov jump process ML estimator. This was done by Albert[4] in a paper where the transition probabilities are derived from estimated transition intensities for the case of stationary intensities. It is shown that if there is a positive probability for the process to be in any of the possible states for a certain time interval, $[0, t)$, the estimator will be consistent and the residuals of the estimated transition intensities will follow an asymptotic normal distribution. These results are true for the case when the number of observations, or the time interval, tends to infinity.

Markov models were used in the social sciences from 1964 and this led to interest towards the estimation of Markov process parameters[104]. Markov processes in the social sciences involved the study of, for example, the movement of people through “Single”, “Married”, “Divorced” states. Tuma is one of the authors from the social sciences who contributed towards the use and implementation of Markov processes. In 1979 Tuma was the lead author of two papers[103,104] that described the likelihood function of a Markov jump process. It is shown in Section 3.4 that the likelihood function in the one paper by Tuma is the same as that of Albert[4] in the case of a stationary Markov process. Tuma does provide an alternative method for deriving the likelihood function which is flexible enough to incorporate nonstationary Markov processes and the effect of covariates on transition rates.

Due to the nature of clinical trial data, the complete outcome of all patients is often not known by the time the trial is finished. Furthermore, since observation times in a clinical trial are often predetermined or sometimes irregular, the exact time at which a process transitions between states can be unknown and some transitions may not be observed at all[56]. This has led to additions to the literature. In 1985 and 1986, Kay[56], and Kalbfleisch and Lawless[53], respectively considered methods for which the assumption of observing transitions exactly, or at equidistant time points were relaxed.

Kalbfleisch and Lawless[53] provide a ML estimator for a Markov process of which all transitions are observed at the same irregular time points. This method requires optimising a likelihood function for which closed form expressions of the ML estimators cannot be provided and therefore they provide an iterative quasi-Newton method to optimise the likelihood function.

The method of Kay[56] does not require all processes to be observed at the same time points and makes provision for the fact that a transition to the death state is often observed within a day and adds this information in the likelihood function. The method of Kay also requires a likelihood function for which closed form expressions of the ML estimators are not available.

There is often great interest in the effect that some characteristics of the units being studied may have on the parameters of a Markov process. As previously discussed, it has been shown that smoking, diabetes and high blood pressure increase the risk of strokes. One may also wish to assess the influence that a medical intervention has on the wellbeing of patients with a certain condition. The methods of Kalbfleisch and Lawless[53] and Kay[56] make provision for the inclusion of covariate effects on the transition parameters.

In 2002, Jackson[49] constructed a library for R called *msm* which can be used to estimate the parameters of Markov processes. Jackson refers to the work by Kalbfleisch and Lawless[53] and Kay[56] for deriving the likelihood function of a Markov jump process which can handle censored data and include covariates. These methods are then used to construct a likelihood function and thereafter estimate the parameters of the process.

It would be difficult to provide an accurate estimate of the number of people using the *msm* package. The *msm* user manual[49] invites the users thereof to inform the author when they use the package for the personal interest of the author. I enquired with the author about the number of people that have informed him of the use of *msm* and he indicated in an email on the 10th of March 2010 that he estimates 100 people have informed him of such usage. It therefore seems that the *msm* package is used frequently. Published examples of the use of the *msm* package include Sweeting et al.[102], Gautrais et al.[37], Grassly et al.[39] and Mekonnen et al.[73]. In an unpublished article by Jackson[50], reference is made to seven other articles that have made use of the *msm* package and it is also mentioned in this article that the *msm* package is “frequently used” in the studies of chronic diseases.

1.4 Summary

The field of pharmacoeconomics is growing in South Africa and more generally the Southern African Development Community[86] and provides a possible approach for optimally allocating scarce financial and human resources to the Southern African population. Due to the use of Markov processes in pharmacoeconomics and the growing demand for pharmacoeconomics, the techniques discussed in this dissertation can be used in the future planning of healthcare resources in Southern Africa.

There has been one Masters[27] and one Doctorate[58] level thesis in South Africa concerning disease modelling but none were focussed on specifically using Markov processes in disease modelling or the estimation of the parameters thereof. This suggests that interest in disease modelling may be growing, but there is still a lack of locally available literature regarding the use of Markov models in disease modelling. Globally, the literature on the estimation of the transition probabilities of Markov chains was initiated by Andersen and Goodman [8] in 1957. This was followed by the extension of these methods to the continuous time Markov jump process transition rates in 1962 by Albert[4]. Thereafter the estimation techniques developed so that censored observations and the effect of covariates on the process parameters could be incorporated.

In this dissertation the background knowledge of Markov processes will be provided from the textbook by Cox et al.[20] and Doob[26]. Then the ML estimators of Andersen and Goodman[8] and Albert[4] will be discussed in detail and implemented with an example. The methods by Tuma et al.[104] will be discussed to indicate an alternative approach to constructing a similar likelihood function to that of Albert.

The *msm* package for R developed by Jackson[49] is thought to be an important and widely used tool in the use of Markov processes in disease modelling and therefore the methodology followed by this package will be explored in detail. The methods by Kalbfleisch and Lawless[53] and also Kay[56] will be discussed since these methods are used in the *msm* package.

The methods by Andersen and Goodman[8], Albert[4], Tuma et al.[104], Kalbfleisch and Lawless[53] and Kay[56] will be implemented and compared with the output from the *msm* package for R produced by Jackson[49].

Twenty one articles were mentioned in Section 1.3.2.1 that used Markov models in disease modelling. These articles were investigated to assess if Markov chain or jump processes were used in the model construction. This could not be determined for two of the articles, ([68] and [29]), due to restrictions on the access to the articles. For the remaining 19 articles, 17 (89%) used Markov jump processes. It therefore seems that time is mostly considered to be continuous when Markov processes are used in disease models. The choice of continuous time over discrete may be caused by the greater ease at which individual based simulation models can be implemented with Markov jump processes compared to Markov chains. Individual based simulation models simulate the path of a few processes individually up to a certain time point or until the process reaches a state from where no transitions can occur. With stationary Markov jump processes, the time until the next event is sampled from a exponential distribution, whereas the the state of the Markov chain needs to be determined at the end of every discrete time point which can result in many transitions to the same state. Methods in which time is considered to be continuous will therefore be discussed in more detail in this dissertation with notes being made on how to calculate the parameters of Markov chain processes from the estimated parameters of Markov jump processes.

Advances in the estimation of the parameters of Markov processes since the development of the *msm* package will be mentioned in Chapter 5 along with Hidden Markov processes, dynamic modelling and Bayesian methods.

Chapter 2: Preliminary background

Chapter 2 will provide background information on some topics being used in the dissertation. Markov processes are a specific type of stochastic process which is a subject that relies on measure theory. We therefore begin the overview with some measure theory results, building up to the definition of a random variable and then a stochastic process. The Markov property for stochastic processes is then discussed with an overview of two types of Markov processes which differ in their treatment of time. Some results for Markov processes are provided followed by a few topics often used in the more general context of survival analysis.

2.1 Measure theory

As in many areas in statistical literature, Markov processes deal extensively with the probabilities of a set of possible results of an experiment whose outcome is unknown before the experiment starts. We are interested in answering questions such as what possible outcomes the experiment can have, what the probabilities are that the process will have such outcomes, and how these probabilities should be measured. This creates a need for some concepts and definitions from measure theory. The definitions introduced were all taken from Cinlar[16] unless stated otherwise.

Definition 1

An experiment whose outcome is not known in advance is defined to be a **random experiment**.

Definition 2

All the possible outcomes of an experiment is defined as the **sample space** and will be denoted by the non-empty set Ω.

In the case of a two sided coin toss experiment, for example, the sample space will be $\{Heads, Tails\}$. Applied to disease modelling, the sample space may be $\{Alive, Dead\}$ which denotes the state of a patient at a specific point in time. The sample space will expand as more states are considered, for example $\{NoTumor, Tumor, Dead\}$.

An event of the sample space is a subset of the sample space and is denoted by A. In the coin toss example a possible event can be $Heads$ or $Tails$ for example. The outcome of an experiment is denoted by ω.

Definition 3

The **complement** of A is denoted by A^C and represents the set $A^C = \{\omega \in \Omega; \omega \notin A\}$.

Definition 4

The **union** of two events A_i and $A_j; i \neq j$ is defined as the set

$$A_i \cup A_j = \{\omega \in \Omega : \omega \in A_i \; or \; \omega \in A_j\}.$$

The union of more than two events is be denoted by $\bigcup_i A_i$.

Definition 5

The **intersection** of two events A_i and $A_j; i \neq j$ is defined as the set

$$A_i \cap A_j = \{\omega \in \Omega; \omega \in A_i \; and \; \omega \in A_j\}.$$

The intersection of more than two events is be denoted by $\bigcap_i A_i$.

Definition 6

Consider a space F and a collection of subsets $A_1, A_2, \ldots \subseteq \Omega$. If $A_i \in F \; \forall i$, F is said to be **closed under the formation of countable unions** if $\bigcup_{i=1}^{\infty} A_i \in F$. Similarly F is said to be **closed under the formation of countable intersections** if $\bigcap_{i=1}^{\infty} A_i \in F$. Furthermore, F is said to be **closed under the formation of complements** if for any subset $A \subseteq \Omega$, $A \in F \Rightarrow A^C \in F$. The notations introduced in this paragraph were taken from Rosenthal[93].

Definition 7

The **empty set** is denoted by ϕ and has the following properties: $\phi = \Omega^{\mathrm{C}}$ and $\Omega = \phi^C$.

Definition 8

Two events A_i and A_j are said to be **disjoint** if $A_i \cap A_j = \phi$.

Definition 9

A **measure** on a space A is defined to be the operator μ such that it has the following properties[43]:

- $\mu(A_i) \geq 0 \ \forall \ A_i \in A$ (nonnegativivity)
- $\mu(\bigcap_i A_i) = \sum_i \mu(A_i)$ if all $A_i's$ are disjoint (countable additivity)

Definition 10

A **counting measure**, C, is defined on any set A, such that $C(A) \subseteq N$ is the number of elements in the set A and N is the set of positive integers. C has the following properties:

- If A is finite we have $C(A) = card(A)$ where $card(A)$ is the number of elements in A.
- If A is infinite we have $C(A) = \infty$

Definition 11

A **probability measure** of a sample space is denoted by the function P and has the following properties:

- $0 \leq P(A) \leq 1$ for any event A
- $P(\Omega) = 1$
- $P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$ for a sequence of disjoint events $A_1, A_2, \ldots$

The definition of a probability measure is taken from Rosenthal[93].

Definition 12

A **sigma algebra** (written as σ-algebra), also known as a σ-field, is the set F which contains Ω and ϕ and is closed under the formation of complements, countable unions and countable intersections.

Definition 13

A **probability space**, also known as a probability triple, is denoted by (Ω, F, P) where

- Ω is the sample space
- F is a σ-algebra
- P is a probability measure which maps $F \rightarrow [0,1]$.

Definition 14

There exists a probability space (Ω, F, P) such that $\Omega = [0,1]$, F contains all the intervals in $[0,1]$ and for any interval $I \in [0,1]$, $P(I)$ is the length of the interval and is as such a **Lebesgue measure** for the space.

Definition 13 and Definition 14 were taken from Rosenthal[93].

Definition 15

A property, say $\mathfrak{D}$, on a measure space (Ω, F, P) is said to hold **almost everywhere** if the property $\mathfrak{D}$ holds for all points in the set, except a set of points for which the measure is zero.

The term "almost everywhere" is also often written as "almost all".

Two properties of probability measures that will be used in this dissertation are provided below:

- The conditional probability of event A given that event B has occurred is denoted by $P(A|B)$ and is calculated as follows:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \text{ given that } P(B) > 0. \qquad (1)$$

- Two events A and B are said to be independent if $P(A \cap B) = P(A)P(B)$. From Equation (1), we then have that if A and B are independent, then $P(A|B) = P(A)$.

Definition 16

A **random variable**, X_t, is a function defined on a probability space (Ω, F, P), which assigns a value to each outcome $\omega \in \Omega$.

Random variables are also called stochastic variables and are written in upper case and the realisation thereof is written in lower case letters, for example x_t. Random variables do not have to be written with a subscript as we do here by adding a t to the notation. The subscript is merely added to illustrate that we talk about the state of a system at a certain point in time. Consider for example a disease modelling example with sample space $\{NoTumour, Tumour, Dead\}$. Before a patient is observed it is not known in which one of these three states the patients will be at certain times. This is therefore a random experiment. A possible outcome for the random variable at various values of t is as follows

$$X_1(\omega) = NoTumor$$

$$X_2(\omega) = NoTumor$$

$$X_3(\omega) = Tumor$$

$$X_4(\omega) = Dead.$$

Definition 17

A **stochastic process** is defined as the collection $\{X_t; t \in T\}$. The set T denotes the **parameter space** of the process. The parameter space can be any indexing parameter but time is usually used being either discrete or continuous. The exhaustive set of all possible values that the stochastic variable can take on is called the **state space** of the process and is denoted by $\mathcal{W}_0$.

If the state space is enumerable, the process is said to have a discrete state space and otherwise the state space is said to be continuous. Only stochastic processes with discrete state spaces and a finite number of states will be considered in this dissertation. A discrete state space consist of $\mathscr{s}$ mutually exclusive and exhaustive states and so we have $\mathcal{W}_0 = \{1, \dots, \mathscr{s}\}$. The definitions of a random variable and a stochastic process of Cinlar[16] have been adopted to suit the purposes of this dissertation.

An observed stochastic process, $\{x_{t_1}, x_{t_2}, \dots, x_{t_n}\}$, can be represented graphically as a sample function. Since we are only concerned with Markov processes with discrete states, a sample function will be discontinuous where jumps occur and will therefore be represented by a step function. An example of a sample function of a Markov jump process with $\mathscr{s} = 10$ is shown in Figure 2.

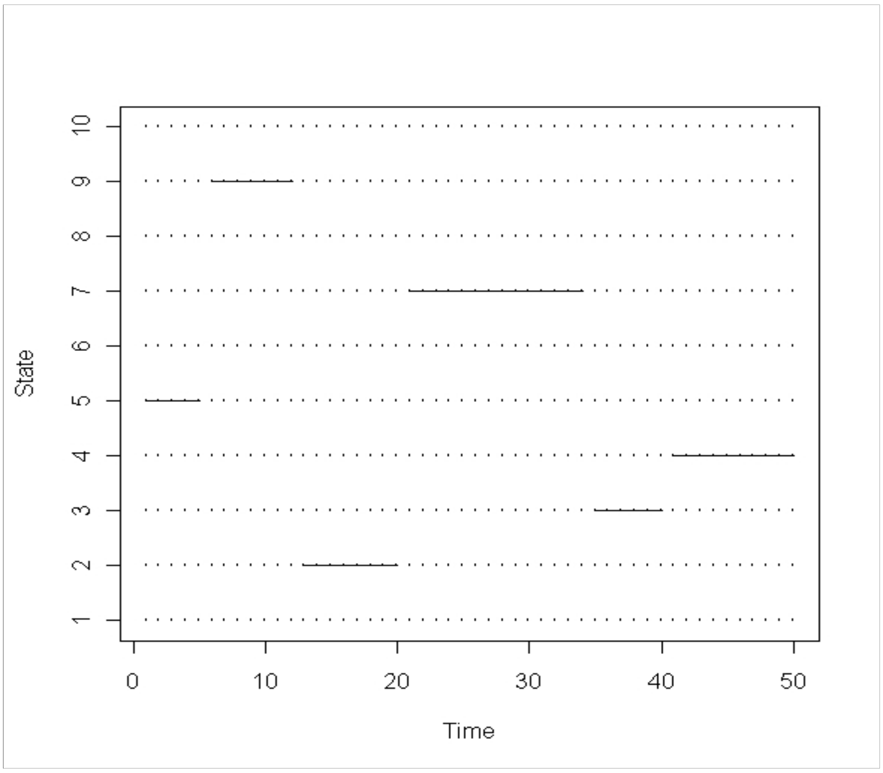

Figure 2: Example of a possible sample function for a Markov jump process

Through the probability space on which the stochastic process is defined, the joint distribution function of a finite number of X_t's can be determined. For any collection from the parameter space, $\{t_1, t_2, \dots, t_n\}$, the joint distribution function of a set of possible outcomes, $\{x_{t_1}, x_{t_2}, \dots, x_{t_n}\}$, of X_t is given by

$$\begin{aligned} F_{X_{t_1},X_{t_2},\dots,X_{t_n}}(x_1, x_2, \dots, x_n) &= P\left[X_{t_1} \le x_1, X_{t_2} \le x_2, \dots, X_{t_n} \le x_n\right] \\ &= P\left[\omega \middle| X_{t_1}(\omega) \le x_1, X_{t_2}(\omega) \le x_2, \dots, X_{t_n}(\omega) \le x_n\right]. \end{aligned}$$

Definition 18

A stochastic process is said to be **stationary**, if for any integer n, and any values $t_1 < t_2 < \cdots < t_n \in T$ and any value k such that $t_1 + k < t_2 + k < \cdots < t_n + k \in T$ the joint distribution function of $X_{t_1}, X_{t_2}, \dots, X_{t_n}$ is the same as the joint distribution function of $X_{t_1+k}, X_{t_2+k}, \dots, X_{t_n+k}$.

2.2 Markov processes

Definition 19

A stochastic process $\{X_t; t \in T\}$ defined on (Ω, F, P) is called a **Markov process** if and only if the following property holds[20]:

$$P\left[X_{t_{n+1}} = x_{t_{n+1}} | X_{t_n} = x_{t_n}, X_{t_{n-1}} = x_{t_{n-1}}, X_{t_{n-2}} = x_{t_{n-2}}, \dots\right] = P\left[X_{t_{n+1}} = x_{t_{n+1}} | X_{t_n} = x_{t_n}\right]. \tag{2}$$

In laymen's terms this means that at time n the probabilities for a state of the process at the next time step, $n+1$, is only dependent on the current state of the process.

Markov processes were introduced in 1907 by the Russian mathematician Andrei Andreivich Markov when he investigated the alternative use of vowels and consonants in the poem Onegin by Poeshkin. He constructed a model where successive results depended on all their predecessors only through the immediate predecessor. The model allowed him to obtain good estimates of the relative frequency of vowels in the poem[36]. Markov processes have received a significant amount of attention since then [70]. The properties of Markov processes are well known and provide a powerful set of results, especially for stationary processes.

Some properties of Markov processes are discussed below. This does not serve as an exhaustive description of the vast amount of properties that are available in the literature but merely focuses on some of the properties that are useful in disease modelling. For more information on the properties of Markov processes and the proofs thereof, the interested reader is referred to Cox[20], Doob[26], Cinlar[16] and Feller[33]. Markov processes will be discussed separately for those with a discrete time parameter (Markov chains) and those processes with a continuous time parameter (Markov jump processes). It should be noted that Equation (2) holds for Markov chains and Markov jump processes.

2.2.1 Markov chains

A brief introduction of the properties of Markov chains will be provided below, all of which are taken from the textbook by Cox[20] unless stated otherwise. For all the discussions on Markov chains the stochastic process $\{X_{\mathscr{t}}; \mathscr{t} \in T\}$ defined on (Ω, F, P) will be considered.

The notation of Markov processes can be simplified by the introduction of a TPM. If we let $P[X_n = j|X_m = i]$ be denoted by $p_{ij}^{(m,n)}$, then the TPM of a stochastic process is the $\mathscr{s} \times \mathscr{s}$ matrix $\boldsymbol{P}^{(m,n)}$ such that $p_{ij}^{(m,n)}$ is the $(i,j)^{\text{th}}$ element of the matrix and $\boldsymbol{P}^{(m,m)}$ is the identity matrix. A TPM is a stochastic matrix with the properties that all the elements are nonnegative and less than or equal to one and that the sum of the elements in every row is one.

<u>Definition 20</u>

> The $1 \times \mathscr{s}$ row vector $\boldsymbol{\rho}_j' = (\rho_1, \rho_2, \dots, \rho_{\mathscr{s}})$ is called a **distribution vector** and its k^{th} element denotes the probability that the process will be in state k at time j and this is also known as **state prevalence**. The distribution vector $\boldsymbol{\rho}_j'$ has the property that its elements sum to one since the process must be in one of the states of the process at any given point in time.

Due to the definitions of the TPM and the distribution vector the following useful relation is obtained:

$$\boldsymbol{\rho}_{j+1}' = \boldsymbol{\rho}_j' \boldsymbol{P}^{(j,j+1)}.$$

To calculate the probability of moving from one state of a Markov process to another between two predetermined time points, the Chapman-Kolmogorov Equations can be used. The Chapman-Kolmogorov equations state:

$$p_{ij}^{(m,n)} = \sum_{k \in \mathcal{W}_0} p_{ik}^{(m,l)} p_{kj}^{(l,n)} \; for\; m \leq l \leq n. \tag{3}$$

In matrix notation this is written as

$$\boldsymbol{P}^{(m,n)} = \boldsymbol{P}^{(m,l)} \boldsymbol{P}^{(l,n)} \; for\; m \leq l \leq n. \tag{4}$$

In the case of a stationary Markov chain, $p_{ij}^{(m,n)}$ is only determined by the difference between m and n. In this case the notation for the one step TPM is simplified to $\boldsymbol{P}$ and the Chapman-

Kolmogorov Equations simplify to $\boldsymbol{P}^{(m,n)} = \boldsymbol{P}^{l-m}\boldsymbol{P}^{n-l} = \boldsymbol{P}^{n-m}$. Given, an initial distribution vector $\boldsymbol{\rho}'_0$, the distribution vector at any time point $k \in T$ can then be calculated with the following formula:

$$\boldsymbol{\rho}'_k = \boldsymbol{\rho}'_0\boldsymbol{P}^k. \tag{5}$$

Equation (5) can be generalised for a nonstationary process as follows:

$$\boldsymbol{\rho}'_k = \boldsymbol{\rho}'_0\boldsymbol{P}^{(0,1)}\boldsymbol{P}^{(1,2)}\boldsymbol{P}^{(2,3)} \dots \boldsymbol{P}^{(k-1,k)}. \tag{6}$$

It should be noted that Equation (5) can be used if k is not an integer. An example of this is when a Markov chain represents yearly transitions between states and the distribution vector needs to be calculated for a time point between the start and end of a year. This can be done if $\boldsymbol{P}$ is decomposed as $\boldsymbol{P} = \boldsymbol{A\Lambda A'}$ where the i^{th} column of $\boldsymbol{A}$ is the i^{th} eigenvector of $\boldsymbol{P}$ and $\boldsymbol{\Lambda}$ is a diagonal matrix with i^{th} entry equal to the i^{th} eigenvalue of $\boldsymbol{P}$. The matrix $\boldsymbol{P}^k$ can then be calculated for any real value of k as follows:

$$\boldsymbol{P}^k = \boldsymbol{A\Lambda}^k\boldsymbol{A}'. \tag{7}$$

$\boldsymbol{\Lambda}^k$ in Equation (7) is calculated by raising each diagonal entry of $\boldsymbol{\Lambda}$ to the power $k \in [0, \infty)$ since the off-diagonal entries of $\boldsymbol{\Lambda}$ are zero.

Equation (7) can then be used to calculate Equation (5) for any real value of k. It should be noted that Equation (5) will still hold if the distribution vector, $\boldsymbol{\rho}'_k$, contains integers representing the absolute number of units in each of the states. In this case the sum of the elements of the distribution vector will equal the number of units being studied and it will be the same value for all values of k.

States can be classified in terms of their accessibility to each other by the transition probabilities to move between them.

Definition 21

State j is said to be **accessible** from state i if there exists some value $n > 0$ such that $p_{ij}^{(n)} > 0$.

Definition 22

If there exists some values $n > 0$ and $m > 0$ such that $p_{ij}^{(n)} > 0$ and $p_{ji}^{(m)} > 0$, states i and j are said to **communicate**.

In laymen's terms state j is said to be accessible from state i if it is possible for the process to move from state i to state j in a finite number of steps and these states communicate if it is possible to move back and forth between these states at least once in a finite number of steps.

Definition 23

Two, or more, states are said to be in the same **equivalence class** if the states communicate with each other.

The states of a stochastic process can therefore be classified into equivalence classes depending on which states communicate with each other.

Definition 24

If there is only one equivalence class, all the states communicate with each other and the stochastic process is said to be **irreducible**.

Let $f_{ij}^{(n)}$ denote the probability that the process will visit state j for the first time after n steps given that the process started in state i and let $f_{ij} = \sum_{n=1}^{\infty} f_{ij}^{(n)}$.

Definition 25

State i is called **recurrent** if $f_{ii} = 1$ and **transient** if $f_{ii} < 1$.

In other words as time tends to infinity, once the process has left a transient state there is a positive probability that the process will not return to the transient state, whereas the process is expected to return to recurrent states at some point in time.

Definition 26

States for which the probability of leaving the state is zero are called **absorbing states**.

Markov processes that model diseases will often have "Death" as a recurrent and absorbing state. If a Markov chain has one recurrent state, the state will be an absorbing state.

For a stationary Markov chain with recurrent and transient states, there is a well known result that can be used to calculate the estimated amount of time that the process will spend in each of the transient states before the process is in one of the recurrent states. This is useful in disease modelling since it can be used to estimate the amount of time that patients are in "Alive" states before moving to the "Death" state.

For a stationary Markov chain with m transient states and $m - s$ recurrent states, the one step TPM can be written as follows:

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{R} \\ \boldsymbol{0} & \boldsymbol{P_1} \end{bmatrix}. \tag{8}$$

In Equation (8), $\boldsymbol{Q}$ refers to the $m \times m$ TPM for the transient states, $\boldsymbol{R}$ refers to the $m \times (m - s)$ TPM for movement from the transient states to the recurrent states, the matrix $\boldsymbol{0}$ is a $(m - s) \times$ m matrix with all elements equal to zero which represent the zero probabilities to move from recurrent to transient states, and $\boldsymbol{P_1}$ represents the $(m - s) \times (m - s)$ TPM for movements between the recurrent states. The following property of $\boldsymbol{Q}$ is useful in disease models:

> The expected number of visits from transient state i to transient state j before moving to a recurrent state is equal to the $(i, j)^{\text{th}}$ element of $(\boldsymbol{I} - \boldsymbol{Q})^{-1}$. (9)

In Equation (9), the expected number of visits to a state can also be interpreted as the amount of time spent in the state.

<u>Definition 27</u>

> The expected number of time units that the process will be in a state in a single visit is called the **mean sojourn time** of state i.

The sum of the i^{th} row of $(\boldsymbol{I} - \boldsymbol{Q})^{-1}$ is the expected amount of time for which the process remains in transient states before moving to a recurrent state if the system starts in the i^{th} transient state.

For example, if we have a disease that can be approximated by a stationary Markov chain with sample space

$$\{DiseaseFree, BeningTumour, MalignantTumour, Metastasis, Death\}$$

and TPM

$$P = \begin{bmatrix} 0.7 & 0.2 & 0.05 & 0 & 0.05 \\ 0.6 & 0.1 & 0.1 & 0.05 & 0.15 \\ 0.05 & 0 & 0.4 & 0.25 & 0.3 \\ 0.05 & 0 & 0 & 0.5 & 0.45 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{10}$$

we then have that

$$Q = \begin{bmatrix} 0.7 & 0.2 & 0.05 & 0 \\ 0.6 & 0.1 & 0.1 & 0.05 \\ 0.05 & 0 & 0.4 & 0.25 \\ 0.05 & 0 & 0 & 0.5 \end{bmatrix},$$

and so

$$(I - Q)^{-1} = \begin{bmatrix} 6.38 & 1.42 & 0.78 & 0.52 \\ 4.39 & 2.09 & 0.71 & 0.57 \\ 0.80 & 0.18 & 1.76 & 0.90 \\ 0.64 & 0.14 & 0.77 & 2.05 \end{bmatrix}. \tag{11}$$

The sum of the rows of the matrix in Equation (11) are given by

$$\sum_{j=1}^{4} (I - Q)^{-1}_{[i,j]} = \begin{bmatrix} 9.10 \\ 7.75 \\ 3.64 \\ 2.91 \end{bmatrix}. \tag{12}$$

So if a patient enters the model in the $DiseaseFree$ state, he/she is expected to stay in the $DiseaseFree$ state for 6.38 time units, be in the $BenignTumour$ state for 1.42 time units, be in the $MalignantTumour$ state for 0.78 time units and be in the $Metastasis$ state for 0.52 time units. After all these visits to the transient states, the patients is expected to move to the death state. It is therefore expected that such a patient will be alive for 9.10 time units from entering the model in the $DiseaseFree$ state. Similarly, a patient entering the model in the $BenignTumour$ state, is expected to live for 7.75 time units. The life expectancy for patients entering the model in the $MalignantTumour$ or $Metastasis$ follow similarly from Equation (12).

Another useful application of $(\boldsymbol{I} - \boldsymbol{Q})^{-1}$ is the ease at which the total cost of treatment before death can be calculated if the cost per cycle of being in each of the transient states is known.

For a patient entering the model in state i, the total cost of treatment can be estimated with $\sum_{j=1}^{m} c_j(\boldsymbol{I}-\boldsymbol{Q})^{-1}{}_{ij}$ where c_j is the cost per time unit spent in state j and $(\boldsymbol{I}-\boldsymbol{Q})^{-1}{}_{ij}$ is the $(i,j)^{\text{th}}$ element of $(\boldsymbol{I}-\boldsymbol{Q})^{-1}$. Similar methods can be used if an estimate of the quality of life of patients is known in each state to calculate the life expectancy of patients weighed by the quality of life of the patients in each state.

There are some disease models in which none of the states communicate and all states, except the absorbing death state, are transient. Such models are called progressive models and apply to diseases where patients are in states from which they cannot be cured and the only possible movement is to for the patient to stay in the same state or move to a state of more severe disease. Examples of such models include Sweeting[102], Klotz[57] and Perez-Ocon[84]. It should be noted that $(\boldsymbol{I}-\boldsymbol{Q})^{-1}$ can still be used for calculating the total number of time units a patient is alive, and therefore the total cost before death, for such diseases if the Q matrix is constructed appropriately. If the $\boldsymbol{Q}$ matrix is the sub matrix of the TPM such that it contains the transition probabilities of moving between all the "alive" states, Equation (8) will still hold since $\boldsymbol{Q}$ will be a upper triangular matrix for which the diagonal elements will be less than 1 (since there will always be a nonzero probability to die and P is a stochastic matrix) and therefore $(\boldsymbol{I}-\boldsymbol{Q})^{-1}$ will exist. This is explained by altering the process considered above. Assume again that the states of a Markov process are

$$\{DiseaseFree, BeningTumour, MalignantTumour, Metastasis, Death\}.$$

These states are assumed to be ordered from the least severe disease state to the most severe disease state followed by death. Assume we have the following TPM:

$$P = \begin{bmatrix} 0.6 & 0.2 & 0.05 & 0.1 & 0.05 \\ 0 & 0.7 & 0.1 & 0.05 & 0.15 \\ 0 & 0 & 0.45 & 0.25 & 0.3 \\ 0 & 0 & 0 & 0.65 & 0.45 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{13}$$

The TPM in Equation (13) implies that patients cannot move back to states of less severe disease and this is illustrated by the upper triangular form of the matrix. The $\boldsymbol{Q}$ matrix with transition probabilities between transient states from Equation (13) is as follows:

$$Q = \begin{bmatrix} 0.6 & 0.2 & 0.05 & 0.1 \\ 0 & 0.7 & 0.1 & 0.05 \\ 0 & 0 & 0.45 & 0.25 \\ 0 & 0 & 0 & 0.65 \end{bmatrix},$$

and so

$$(I - Q)^{-1} = \begin{bmatrix} 2.5 & 1.67 & 0.33 & 1.19 \\ 0 & 3.33 & 0.61 & 0.91 \\ 0 & 0 & 1.82 & 1.30 \\ 0 & 0 & 0 & 2.86 \end{bmatrix}. \tag{14}$$

The sum of the rows of the matrix in Equation (14) are given by

$$\sum_{j=1}^{4} (I - Q)^{-1}_{[i,j]} = \begin{bmatrix} 5.68 \\ 4.85 \\ 3.11 \\ 2.86 \end{bmatrix}. \tag{15}$$

We now see that patients that enter the process in the $BeningTumour$ state are expected to spend 0 time units in the $DiseaseFree$ state as expected. Patients from the Markov chain defined by Equation (13) are expected to live 5.68 time units when entering from the $DiseaseFree$ state which is less than the life expectancy of patients entering the $DiseaseFree$ state in the Markov chain represented by Equation (10). If we compare Equations (12) and (15), we see that the progressive model described by Equation (13) is associated with a more severe disease than that of model described by Equation (10) due to a lower life expectancy in all states.

2.2.2 Markov jump processes

As previously mentioned, Markov jump processes are a specific type of Markov process whereby the time parameter is continuous but with the understanding that the property stated in Equation (2) still holds. Some of the properties of Markov jump processes will be discussed below; all of which are taken from the textbook by Doob[26] unless stated otherwise.

A stochastic process $\{X_t; 0 \leq t < \infty\}$ defined on the probability triple (Ω, F, P) will be considered for all the discussions on Markov jump processes.

The transition probabilities of Markov jump processes can also be written in terms of a TPM so that $p_{ij}(s,t) = P[X_t = j | X_s = i]$ is the $(i,j)^{\text{th}}$ entry of $\boldsymbol{P}^{(s,t)}$. $\boldsymbol{P}^{(s,s)}$ is understood to be the identity matrix.

The Chapman-Kolmogorov equations are still valid for the Markov jump process and so Equation (4) also holds for Markov jump processes.

Due to the continuous time nature of Markov jump processes, it is more convenient to think of time in very small intervals.

Definition 28

By making the time intervals small enough, we work with the rate of change of the transition probability and thereby define the **transition rate** (also known as transition intensity) at time t as follows:

$$\begin{aligned}\alpha_{ij}(t) &= \lim_{\Delta t\to 0}\frac{p_{ij}(t,t+\Delta t)-p_{ij}(t,t)}{\Delta t}\\ &= \begin{cases}\lim\limits_{\Delta t\to 0}\dfrac{p_{ij}(t,t+\Delta t)}{\Delta t} & if\ j\neq i\\ \lim\limits_{\Delta t\to 0}\dfrac{p_{ij}(t,t+\Delta t)-1}{\Delta t} & if\ j=i\end{cases}.\end{aligned} \tag{16}$$

For $i \neq j, p_{ij}(t,t)$ in Equation (16) is equal to zero since the probability to move between two states is zero if no time has elapsed.

Let $\boldsymbol{A}(t)$ be the matrix such that the $(i,j)^{\text{th}}$ element is $\alpha_{ij}(t)$. The matrix $\boldsymbol{A}(t)$ is called the transition rate, or transition intensity, matrix. The diagonal entries of the transition rate matrix have the following property:

$$\alpha_{ii}(t) = -\sum_{\substack{k\in W_0\\ k\neq i}}\alpha_{ik}(t). \tag{17}$$

Let $\alpha_i(t) = \lim\limits_{\Delta t\to 0}\frac{1-p_{ii}(t,t+\Delta t)}{h} = \sum_{\substack{k\in w_0\\ k\neq i}}\alpha_{ik}(t)$.

We can write the following properties of the TPM and the transition rate matrices:

$$\frac{\partial}{\partial t}\boldsymbol{P}^{(t,t+\Delta t)} = \boldsymbol{P}^{(t,t+\Delta t)}\boldsymbol{A}(t) \tag{18}$$

$$\frac{\partial}{\partial t}\boldsymbol{P}^{(t,t+\Delta t)} = -\boldsymbol{A}(t)\boldsymbol{P}^{(t,t+\Delta t)} \tag{19}$$

Equations (18) and (19) are called the Kolmogorov forward and Kolmogorov backward equations respectively.

2.2.2.1 Some results for stationary processes

For the case of a stationary process we have

$$\begin{aligned}\alpha_{ij}(t) &= \lim_{\Delta t\to 0}\frac{p_{ij}(t,t+\Delta t)-p_{ij}(t,t)}{\Delta t}\\ &= \lim_{\Delta t\to 0}\frac{p_{ij}(0,\Delta t)-p_{ij}(0,0)}{\Delta t}\\ &= \lim_{\Delta t\to 0}\frac{p_{ij}(\Delta t)-p_{ij}(0)}{\Delta t}\\ &= \alpha_{ij},\text{ say}\end{aligned}$$

and so we simplify the notation such that the transition rate matrix of a stationary process is written as $\boldsymbol{A}$.

In an attempt to solve the differential equation in Equation (18) we define the following matrix function:

$$e^{\boldsymbol{X}} = \sum_{i=0}^{\infty}\frac{\boldsymbol{X}^i}{i!}\quad with\ \boldsymbol{X}^0 = \boldsymbol{I}. \tag{20}$$

Equation (20) can also be written conveniently as:

$$e^{(t-s)\boldsymbol{X}} = \sum_{i=0}^{\infty}\frac{\boldsymbol{X}^i}{i!}(t-s)^i. \tag{21}$$

If we take the derivative of Equation (21) with respect to t we get the following:

$$\begin{aligned}\frac{\partial}{\partial t}e^{(t-s)\boldsymbol{X}} &= \frac{\partial}{\partial t}\sum_{\mathrm{i}=0}^{\infty}\frac{\mathbf{X}^{\mathrm{i}}}{\mathrm{i}!}(\mathrm{t}-\mathrm{s})^{\mathrm{i}} \\ &= \sum_{i=0}^{\infty}\frac{\boldsymbol{X}^{i}}{i!}i(t-s)^{i-1} \\ &= \sum_{i=1}^{\infty}\frac{\boldsymbol{X}^{i}}{(i-1)!}(t-s)^{i-1} \\ &= \sum_{i=1}^{\infty}\frac{\boldsymbol{X}^{i-1}}{(i-1)!}(t-s)^{i-1}\boldsymbol{X} \\ &= \sum_{i=0}^{\infty}\frac{\boldsymbol{X}^{i}}{i!}(t-s)^{i}\boldsymbol{X} \\ &= e^{(t-s)\boldsymbol{X}}\boldsymbol{X}. \end{aligned} \tag{22}$$

We assume that the differential operator in Equation (22) can be taken into the infinite sum. Equation (22) indicates that

$$\boldsymbol{P}^{(s,t)} = e^{(t-s)\boldsymbol{A}} \tag{23}$$

is a solution to the Kolmogorov forward equations in the case where the process is stationary. We therefore have the following equation for calculating the TPM for movements in t time units from the transition intensity matrix:

$$\boldsymbol{P}^{(t)} = e^{\boldsymbol{A}t} \tag{24}$$

where the exponential matrix function is defined in Equation (21).

Equation (23) can be used to estimate the TPM of a Markov chain based on the estimation of the transition intensity matrix of a Markov jump process for any discretization of time.

The following property of the relationship shown in Equation (24) is useful:

$$\begin{aligned}\boldsymbol{P}^{(t)} &= \sum_{i=0}^{\infty}\frac{\boldsymbol{A}^i t^i}{i!}\\ &= I + \boldsymbol{A}t + \frac{\boldsymbol{A}^2t^2}{2!} + \frac{\boldsymbol{A}^3t^3}{3!} + \cdots\\ &= \boldsymbol{BIB}^{-1} + \boldsymbol{B}diag(d_1,\dots,d_s)\boldsymbol{B}^{-1}\frac{t}{1!} + \boldsymbol{B}diag\left(d_1{}^2,\dots,d_s{}^2\right)\boldsymbol{B}^{-1}\frac{t^2}{2!}\\ &\qquad + \boldsymbol{B}diag\left(d_1{}^3,\dots,d_s{}^3\right)\boldsymbol{B}^{-1}\frac{t^3}{3!} + \cdots\end{aligned}$$

where $d_1,\dots,d_S$ are the eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$ is a matrix of which the i^{th} column is the eigenvector associated with the i^{th} eigenvalue d_i of $\boldsymbol{A}$. Now, we have:

$$\begin{aligned}\boldsymbol{P}^{(t)} &= \boldsymbol{B}\left[\sum_{i=0}^{\infty} diag\left(\frac{d_1^i t^i}{i!},\dots,\frac{d_S^i t^i}{i!}\right)\right]\boldsymbol{B}^{-1}\\ &= \boldsymbol{B}\left[diag\left(\sum_{i=0}^{\infty}\frac{d_1^i t^i}{i!},\dots,\sum_{i=0}^{\infty}\frac{d_S^i t^i}{i!}\right)\right]\boldsymbol{B}^{-1}\\ &= \boldsymbol{B}[diag(e^{d_1t},\dots,e^{d_St})]\boldsymbol{B}^{-1}.\end{aligned} \tag{25}$$

The property shown in Equation (25) is stated in the article by Kalbfleisch and Lawless[53] and derived in this dissertation for completeness.

Doob[26] proves the following three properties for a separable[1] stationary Markov jump process which will be used in deriving the ML estimator of a stationary Markov jump process.

1. If the process starts in a certain state at time t_0, the probability that the process will stay in that state for τ time units is as follows:

$$P\left[X_t = i, t_0 \le t \le t_0 + \tau \mid X_{t_0} = i\right] = e^{-\alpha_i\tau}. \tag{26}$$

   The time to transition out of a state i therefore has an exponential distribution with parameter α_i.

[1] The interested reader is referred to page 50-51 of Doob[26] for a definition of a separable Markov process. All Markov processes will be assumed to be separable for the purposes of this dissertation.

2. Consider the separable Markov jump process with initial state $X_{t_0} = i$ and $\alpha_i > 0$. A sample function discontinuity exists with probability one for some $\tau > t_0$. If we know the sample discontinuity is in the interval $[t_0, t_0 + \tau)$, then we have

$$P\left[X_{t_0+\tau} = j \middle| X_{t_0} = i\right] = \frac{\alpha_{ij}}{\alpha_i}. \tag{27}$$

3. Almost all sample functions are step functions with a finite number of jumps in any finite time interval. (28)

2.3 Basics of survival analysis and terminology

Survival analysis, in laymen's terms, consists of the study of the time until a specific event occurs. Examples include the study of the time until manufactured units fail to perform a specific duty, the time until death of a human or animal, the time until a person is involved in a car crash leading to an insurance claim or the time until a person's health deteriorates to a specific state of worse health. This dissertation is focussed on health applications of survival analysis, but many of the concepts and techniques introduced have applications in other sciences.

Only a short overview of some of the terminology and concepts in survival analysis is provided. These concepts and definitions are all based on the book by Collet[18]. As mentioned by Collet a more detailed account of survival analysis can be found in the book *The Statistical Analysis of Failure Time Data* written by Kalbfleisch and Prentice, published in 1980[55].

Survival data often consists of a sample of research units thought to be representative of the population under study for which the time until a specific event occurs is measured. In medical applications this is done by recruiting patients during a recruitment period and then studying the patient until the event under study occurs. This is known as a clinical trial. Due to practical and financial constraints clinical trials are planned to stop at a specific date. It is therefore possible that the event under study has not occurred in some patients by the time the study ends. The time until event is therefore censored for such patients. There are two types of censoring which are used in this dissertation and these terms are defined below.

Definition 29

Right censoring occurs when the time until the event is not measured, but it is known to be at least a certain amount.

In clinical trials, right censoring typically occurs when the time until death is measured and patients are still alive by the time the study ends. Patients may also chose to withdraw from a clinical trial before the event under study is measured.

Definition 30

Interval censoring occurs when the time to an event is not known exactly, but it is known to be within a specific time interval.

Interval censoring can for example occur if the time until cancer recurrence is studied and a patient is monitored at the start of the study and one year later. If it is known that the cancer has not recurred at the start of the study, and it has recurred by the end of the first year, but the exact date of recurrence is unknown, the recurrence time is known to be between the start of the study and one year later and is therefore interval censored.

In survival analysis the time to an event is a stochastic variable. For the purposes of this introductory section on survival analysis we denote the time to event stochastic variable as T and the realisation thereof as t. If we let the probability distribution function of T be denoted by $f(t)$, we have the following cumulative distribution function for T:

$$F(t) = P[T \leq t] = \int_0^t f(u)du.$$

$F(t)$ can therefore be interpreted as the probability that the time to an event is at most t.

Definition 31

The **survival function**, $S(t)$, is defined as the probability that the event does not occur by time t. The survival function is therefore equal to $P[T > t] = 1 - F(t)$.

Definition 32

The **hazard function**, $h(t)$, is defined to be the limiting probability that an event will happen between time t and time $t + \Delta t$, given that the event has not happened up until time t, divided by Δt. The hazard function is defined to be

$$\begin{aligned} h(t) &= \lim_{\Delta t \to 0} \frac{P[t \leq T < t + \Delta t | t \geq T]}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{P[(t \leq T < t + \Delta t) \,\&\, (t \geq T)]}{\Delta t} \frac{1}{P[t \geq T]} \\ &= \lim_{\Delta t \to 0} \frac{P[t \leq T < t + \Delta t\ ]}{\Delta t} \frac{1}{P[t \geq T]}. \end{aligned}$$

The hazard function is therefore a rate and not a probability and takes on any value in the interval $[0, \infty)$.

The second step in the above derivation arises from Equation (1). The $\lim_{\Delta t \to 0} \frac{P[t \leq T < t + \Delta t\ ]}{\Delta t}$ is the derivative of $F(t)$ and so we have

$$h(t) = \frac{f(t)}{S(t)} \tag{29}$$

since $P[t \geq T] = S(t)$. Noting that $S'(t) = \frac{d}{dt}\{1 - F(t)\} = -f'(t)$ we see that if we let

$$h(t) = -\frac{d}{dt}\ln\big(S(t)\big) \tag{30}$$

we get

$$h(t) = -\frac{1}{S(t)}\frac{d}{dt}S(t) = \frac{f(t)}{S(t)}$$

which satisfies Equation (29). Equation (30) therefore provides a method for calculating the hazard function if the survival function is known. We therefore also have that

$$f(t) = h(t)S(t). \tag{31}$$

If one consider the time until a person dies, the hazard rate at time t is the limiting probability that the person dies at time $t + \Delta t$ given that the person is still alive at time t divided by Δt, i.e.

$$h(t) = \lim_{\Delta t \to 0} \frac{P[t \leq T < t + \Delta t | t \geq T]}{\Delta t}. \tag{32}$$

For a two state Markov process with "Alive" and "Dead" states, the transition intensity at time t is the derivative of the transition probability from alive to dead at time t, i.e. $\alpha_{Alive\ Dead}(t) = \lim_{\Delta t \to 0} \frac{p_{Alive\ Dead}(t,t+\Delta t)}{\Delta t}$. But $p_{Alive\ Dead}(t, t+\Delta t)$ is the same as $P[t \leq T < t + \Delta t | t \geq T]$ in Equation (32) since $p_{Alive\ Dead}(t, t+\Delta t)$ implies that a person is still alive at time t and will be dead by time $t + \Delta t$. We therefore see that the transition intensity from state i to state j can be interpreted as the hazard rate for the event that will cause the process to move from state i to state j.

Definition 33

The **cumulative hazard function**, $H(t)$, is defined to be

$$H(t) = \int_0^t h(u)du.$$

From Equation (30) we see that $H(t)$ can be written as

$$\begin{aligned} H(t) &= \int_0^t \left\{-\frac{d}{dt} ln\big(S(u)\big)\right\} du \\ &= -ln\big(S(u)\big)\Big|_0^t \\ &= -ln\big(S(t)\big) + ln\big(S(0)\big) \\ &= -ln\big(S(t)\big) + ln(1) \\ &= -ln\big(S(t)\big) \end{aligned}$$

and so $S(t) = e^{-H(t)}$.

A well known concept in survival analysis is the non-parametric Kaplan-Meier (KM) estimate of the survival function. The KM estimate of the survival function is also called the product limit estimator. The KM estimator is briefly described for completeness since it is a well known concept in survival analysis and the interested reader is encouraged to read pages 19 to 31 of Collet[18] for more information.

Suppose n individuals are studied until an event occurs with the following survival times $t_1, t_2, \dots, t_n$ where $r \leq n$ of these survival times are observed exactly and the rest are right censored. Assume it is possible for some individuals to have the same survival time. Let $t_{(1)}, t_{(2)}, \dots, t_{(r)}$ be the r ordered survival times. We denote by n_j the number of individuals for whom it is known that the event has not occurred just before time $t_{(j)}$. Each n_j therefore includes the individual(s) for whom the event is observed at time $t_{(j)}$ but excludes survival times that are right censored and less than $t_{(j)}$. Let d_j denote the number of individuals for whom it is known that the event occurs at time $t_{(j)}$. The KM estimate of the probability of no event up to time t , if $t \in [t_{(k)}, t_{(k+1)})$, is:

$$\hat{S}(t) = \prod_{j=1}^{k} \left(\frac{n_j - d_j}{n_j} \right).$$

The standard error of the KM estimate is estimated as follows:

$$se\left(\hat{S}(t)\right) = \hat{S}(t) \sqrt{\sum_{j=1}^{k} \frac{d_j}{n_j(n_j - d_j)}}.$$

The KM estimate of the hazard function at time t, where $t \in [t_{(k)}, t_{(k+1)})$ is given by:

$$\hat{h}(t) = \frac{d_k}{n_k(t_{(k+1)} - t_{(k)})}.$$

Another well known concept in survival analysis is the Cox proportional hazard model. This is used to determine the effect of characteristics associated with individuals on the hazard rate. One may for example want to determine the risk of smoking and diabetes on strokes as done in the Framingham heart study[23]. The Cox proportional hazard model[19] provides a method for estimating the effect of covariates on the hazard function.

Proportional hazard can be understood as follows. Suppose the hazard function is computed for two groups of individuals, say Group A and Group B. Now we let $h_A(t)$and $h_B(t)$ be the respective hazard functions of the two groups and we write $h_A(t)$and $h_B(t)$ relative to each other in what is known as the proportional hazards model as

$$h_A(t) = \theta h_B(t).$$

where θ is known as the relative hazard or hazard ratio. It is clear that $\theta > 1$ implies $h_A(t) > h_B(t)$ and $\theta < 1$ implies $h_A(t) < h_B(t)$.

The hazard function of Group A can therefore be calculated by the hazard function of Group B multiplied by the relative hazard. This idea is generalised to the Cox proportional hazard model where the hazard function of individual i at time t is written as

$$h_i(t) = h_0(t)e^{(\beta_1 z_{1i}+\beta_2 z_{2i}+\cdots+\beta_p z_{pi})}. \tag{33}$$

where $\{z_{1i}, z_{2i}, \ldots, z_{pi}\}$ are p covariates of the i^{th} individual and $\{\beta_1, \beta_2, \ldots, \beta_p\}$ are the effects of the covariates on the hazard function. Here $h_0(t)$ is known as the baseline hazard and represents the hazard of an individual for which $\beta_1 z_{1i} + \beta_2 z_{2i} + \cdots + \beta_p z_{pi} = 0$.

2.4 Discussion

We provided the definitions of Markov chains and Markov jump processes, $\{X_{\mathcal{t}}; 0 \leq \mathcal{t} < \infty\}$, with the domain of $\mathcal{t}$ being discrete and continuous respectively. We write $X_{\mathcal{t}}$ for the state of the process at time $\mathcal{t}$ and reserve the use of the symbol $\mathcal{t}$ to indicate the time at which the state of the process is considered. We will use the symbol t to indicate the time between events such that t can be calculated by the difference between two times for which the state of the process is considered.

We provided definitions of censoring in Section 2.3 and will illustrate this with data from a heart transplant sample studied by Klotz and Sharpless[57].

In the study by Klotz and Sharpless[57] heart transplant patients were assumed to be disease free after a heart transplant and were invited to be investigated on a yearly basis after the surgery. The disease status of these patients were then categorised based on the extent to which their major vessels narrowed. Patients are categorised as $1 \equiv Disease\ free, 2 \equiv Mild\ disease$ or $3 \equiv Moderate\ to\ severe\ disease$. Patients that died were classified as being in state 4.

Klotz and Sharpless[57] observed 240 patients which were followed up for up to nine years. The studied recorded the state of patients that underwent heart transplant surgery from January 1979 until May 1990. From 1983 it was decided that patients should be recalled two years after the surgery and that those that were disease free after two years should be followed up every second year and patients with some form of disease were invited to come for investigations every year. The observations of the 240 patients are displayed in the article by Klotz and Sharpless[57] and the first 20 observations are shown in Table 2 to illustrate the practical difficulties of dealing with censored data. Table 2 shows the state of patients after the heart transplant, $\mathcal{t}_0$, one year after the transplant, $\mathcal{t}_1$, and so on. For the purposes of discussion the events between $\mathcal{t}_0$ and $\mathcal{t}_1$ will be referred to as occurring in the first year, between t_1 and t_2 as the second year and other references to time following similarly. Table 2 shows that patient 1 was observed every year until death six years after the heart transplant, i.e. death in the seventh year. We know that patient 5 was in state 1 from the surgery until the seventh year and in state 3 in the ninth year. Patient 5 could have been in state 1,2 or 3 in the eight year but the exact state is censored. This is therefore an example of interval censoring. Patient 6 is known to be in the state 3 in the third year and we know the patient died in the fifth year. Since patients can only move to state 4 from state 3 one can assume that patient 6 was in state 3 in the fourth year since time of death is usually observed with accuracy up to one day as noted by Kay[56]. In practice such an assumption should be validated with the data source and Klotz and Sharpless[57] did not mention the accuracy at which death times were recorded. The state of patient 7 was not recorded for the second year but we know that this patient was in state 1 in the first and third years and therefore we know the patient was in state 1 in the second year based on the possible transitions of the Markov process. Thus even though the state of patient 7 was not observed in the second year the true value of the state of the patient is not censored.

There are many other patients in the sample of Klotz and Sharpless[57] but all these cases will not be discussed since the aim was just to indicate the types of censoring one may encounter in practice.

PatientID	t_0	t_l	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
1	1	1	2	2	2	3	4			
2	1	1	3	4						
3	1	1	1	1	1	1	2	3	3	4
4	1	4								
5	1	1	1	1	1	1	1		3	4
6	1	1	3		4					
7	1		1	1	1	1	1	2		
8	1	1	1	1	1	1	1	1	1	
9	1	4								
10	1	1	1	1	2	2	4			
11	1	1	1	1	1	1		1	1	
12	1	1	2	3	4					
13	1	1	1	1	2	2	2	3	3	
14	1	1	1	4						
15	1	1	4							
16	1	1	1	1	1	1	2	2	2	
17	1	4								
18	1	1	1	4						
19	1	1	1	1	2	3	3		3	
20	1	1	1	1	1	1	2	2		

Table 2: Sample of 20 observation of the heart transplant patients[57].
$1 \equiv$ Disease free, $2 \equiv$ Mild disease, $3 \equiv$ Moderate or severe disease, $4 \equiv$ Death

We indicated two methods in Section 2.2.2.1, Equations (24) and (25), to calculate the TPM matrix for a time period based on the transition intensity matrix of a stationary Markov jump process. Equation (24) involves a infinite sum which will be approximated with finite number of terms in practise. The estimated TPMs calculated from a different number of terms in the summation of Equation (24) converge as the number of terms increase. Equation (25) has an advantage over Equation (24) in that one does not have to approximate an infinite sum, but has the potential disadvantage that it can be computationally intensive to calculate the eigenvalues and eigenvectors of a matrix and the inverse of the matrix of eigenvectors. We investigated this by using Equation (24) with a different number of terms in the summation and also Equation (25) to calculate the TPM in one time period of a Markov jump process with transition intensity matrix given as

$$\begin{bmatrix} -3.6 & 0.8 & 1.3 & 0.4 & 0.9 & 0.2 \\ 1.2 & -3.2 & 0.3 & 0.2 & 1.4 & 0.1 \\ 0.3 & 1.4 & -3.1 & 0.7 & 0.5 & 0.2 \\ 0.3 & 0.5 & 1.2 & -3.5 & 0.4 & 1.1 \\ 0.1 & 1.2 & 0.7 & 0.5 & -2.9 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \qquad (34)$$

The structure of the model was chosen so that it has one absorbing state but is more complex than a three state model so that the effect of processes with a bigger state space can be investigated and the entries of Equation (34) were chosen arbitrarily. The estimated TPM and the respective process time is show in Table 3 below. The computation time can be calculated in R using the *proc.time()* function.

Table 3 indicates that Equation (24) produces consistent estimates for the TPM when 20 or more terms are used in the summation. We see that the TPM calculated with 10 terms in the summation of Equation (24) produces an invalid TPM since some of the entries are less than zero which are inconsistent with the properties of probabilities (See Definition 11). The eigenvalues and eigenvectors of the transition intensity matrix in Equation (34) were complex numbers, but due to the nature of Equation (25) it does not produce transition probabilities that are complex numbers. We see that the same TPM was estimated with Equations (24) and (25) when 20 or more terms are used in the summation of Equation (24) and that there is a small difference in calculation time between using Equation (25) compared to Equation (24).

We choose to use Equation (24) with 30 terms in the summation when calculating the TPM from a stationary transition intensity matrix.

Approach for calculating TPM	TPM	Run time (seconds)
Equation (24) with 10 terms	$\begin{bmatrix} 0.0976 & 0.3027 & 0.0993 & 0.1406 & 0.1071 & 0.2527 \\ -0.047 & 0.2407 & 0.3077 & 0.0805 & 0.1835 & 0.2345 \\ 0.2503 & 0.0831 & 0.1344 & 0.0280 & 0.2340 & 0.2702 \\ -0.0099 & 0.2422 & 0.0775 & 0.1805 & 0.0911 & 0.4185 \\ 0.1802 & 0.155 & 0.0657 & 0.0708 & 0.2278 & 0.3005 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	0.01
Equation (24) with 20 terms	$\begin{bmatrix} 0.1151 & 0.1943 & 0.1633 & 0.0888 & 0.1818 & 0.2568 \\ 0.1128 & 0.2163 & 0.1509 & 0.0856 & 0.2020 & 0.2325 \\ 0.0987 & 0.2000 & 0.1724 & 0.0916 & 0.1755 & 0.2618 \\ 0.0728 & 0.1444 & 0.1293 & 0.0898 & 0.1288 & 0.4349 \\ 0.0881 & 0.1890 & 0.1408 & 0.0838 & 0.1986 & 0.2996 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	0.01
Equation (24) with 30 terms	$\begin{bmatrix} 0.1151 & 0.1943 & 0.1633 & 0.0888 & 0.1818 & 0.2568 \\ 0.1128 & 0.2163 & 0.1509 & 0.0856 & 0.2020 & 0.2325 \\ 0.0987 & 0.2000 & 0.1724 & 0.0916 & 0.1755 & 0.2618 \\ 0.0728 & 0.1444 & 0.1293 & 0.0898 & 0.1288 & 0.4349 \\ 0.0881 & 0.1890 & 0.1408 & 0.0838 & 0.1986 & 0.2996 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	0.01
Equation (24) with 40 terms	$\begin{bmatrix} 0.1151 & 0.1943 & 0.1633 & 0.0888 & 0.1818 & 0.2568 \\ 0.1128 & 0.2163 & 0.1509 & 0.0856 & 0.2020 & 0.2325 \\ 0.0987 & 0.2000 & 0.1724 & 0.0916 & 0.1755 & 0.2618 \\ 0.0728 & 0.1444 & 0.1293 & 0.0898 & 0.1288 & 0.4349 \\ 0.0881 & 0.1890 & 0.1408 & 0.0838 & 0.1986 & 0.2996 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	0.01
Equation (25)	$\begin{bmatrix} 0.1151 & 0.1943 & 0.1633 & 0.0888 & 0.1818 & 0.2568 \\ 0.1128 & 0.2163 & 0.1509 & 0.0856 & 0.2020 & 0.2325 \\ 0.0987 & 0.2000 & 0.1724 & 0.0916 & 0.1755 & 0.2618 \\ 0.0728 & 0.1444 & 0.1293 & 0.0898 & 0.1288 & 0.4349 \\ 0.0881 & 0.1890 & 0.1408 & 0.0838 & 0.1986 & 0.2996 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	0.03

Table 3: Comparison of methods for calculating the TPM from the transition intensity matrix

2.5 Summary

We have defined a probability measure that can be used to calculate the probability of a certain event of an experiment. We defined a probability triple which consists of a space of all possible outcomes of an experiment and the measure of the probability of any subset of events in the set of possible outcomes. We defined a random variable and a stochastic process in terms of the probability triple. We discussed the Markov chain and jump processes which are specific types of stochastic processes and provided some results for Markov processes that can be used to calculate the amount of time that patients are expected to live if they have a certain disease, when modelled as a Markov process. Markov chains are defined in terms of the TPM and Markov jump processes are defined in terms of the transition intensity matrix. We derived and compared two methods, Equations (24) and (25), to estimate the TPM from the transition intensity matrix. It was illustrated that these methods provide similar TPMs when more than 30 terms are used in Equation (24) and that Equation (25) is not much more computationally intensive when compared to Equation (24).

Some topics from survival analysis were discussed and the definitions of censoring was explained with reference to a heart transplant study by Klotz and Sharpless[57].

Chapter 3: Maximum likelihood estimation of the parameters of Markov processes

3.1 Introduction

A valuable result has been given in Section 2.2.1 that can be used for calculating the total amount of time that a stationary Markov chain will be in transient states before moving to recurrent states. There is no explicit formula that can be used in the case of a nonstationary Markov chain, but it is possible to simulate the chain with a specific initial distribution vector and then calculating the distribution vector over time and thereby determining the amount of time it will take the process to move through all the transient states. Similar simulation methods can be used, if the parameters of a Markov jump process are known, to determine the amount of time the process spends in transient states before moving to the recurrent state. These methods are however only possible if the TPM is known.

This chapter will discuss ML methods for estimating the TPM of Markov chain and Markov jump processes with reference to the articles by Anderson and Goodman[8] and Albert[4] respectively. The ML estimator of Markov jump process will be further discussed by providing the ML estimator of Tuma et al.[104] which is an alternative, more intuitive, derivation of the ML estimators given by Albert[4]. The possibility of including covariate effects in the estimator of Tuma et al. will also be shown.

The ML estimators of Markov jump processes will then be extended by relaxing the assumptions made on the observation scheme of the processes so that state transitions do not have to be observed exactly. We will discuss the methods by Kalbfleisch and Lawless[53] which assumes all process are observed at the same irregular time points and also the method of Kay[56] for which all observations can be made at possibly different irregular time points. Notes will be made on how covariate effects can be included in these methods. The methods of Kalbfleisch and Lawless[53] and Kay[56] are discussed since they are used in the *msm* package for R developed by Jackson[49] and we would like to explain the methodology of Jackson since this R package is so widely used, as previously mentioned.

3.2 Discrete time: Markov chains

As mentioned in Section 1.3.2.1, the first estimation procedure for the TPM of a Markov process is the 1957 paper by Anderson and Goodman[8]. In this paper a ML estimator for the TPM of a stationary and nonstationary Markov chain is derived. It is assumed that the process is observed at a finite number of equidistant time points and that the process has discrete states. No restrictions are made on the number of equivalence classes of the Markov process.

Suppose that a sample of n Markov chains defined on (Ω, F, P) with s states is observed until time T, i.e. $\{X_1^1, X_2^1, \dots, X_T^1, X_1^2, X_2^2, \dots, X_T^2, \dots, X_1^n, X_2^n, \dots, X_T^n\}$. We denote by $\{X_0^1, X_0^2, \dots, X_0^n\}$ the starting point of each observed process.

From the initial state each process can make s^T possible transitions. The probability of such a sequence of transitions is given by

$$p_{X_0^i X_1^i}(0,1) p_{X_1^i X_2^i}(1,2) \dots p_{X_{T-1}^i X_T^i}(T-1,T) \tag{35}$$

for the i^{th} observed process due to the Chapman-Kolmogorov equation.

Equation (35) simplifies if the process is stationary by writing $p_{X_{t-1}^i X_t^i}(t-1,t) = p_{X_{t-1}^i X_t^i}$.

Now, let $n_{X_0,X_1,X_2,\dots,X_T}$ be the number of processes for which the sequence of states observed in the experiment is $X_0, X_1, X_2, \dots, X_T$. An example of possible observations for a process with $s = 3$ states is shown in Table 4.

Observation ID	Time	State occupied
001	0	1
001	1	1
001	2	2
001	3	1
001	4	3
002	0	1
002	1	3
002	2	3
002	3	1
002	4	2
003	0	1
003	1	1
003	2	2
003	3	1
003	4	3

Table 4: Sample observations

For the sample observations shown in Table 4 we will have $n_{1,1,2,1,3} = 2$ and $n_{1,3,3,1,2} = 1$. Now, let $n_{gj}(t-1,t)$ represent the number of individuals that are in state g at time $t-1$ and in state j at time t (also known as transition counts) and let

$$n_g(t-1,t) = \sum_{j=1}^{\mathcal{S}} n_{gj}(t-1,t) \,\forall\, t = 1, \dots, T$$

and

$$n_{gj} = \sum_{t=1}^{T} n_{gj}(t-1,t).$$

With reference to the sample data shown in Table 4 we will have the following values for the defined statistics:

$n_{1,1}(0{,}1) = 2$	$n_1(0{,}1) = 3$	$n_{1,1} = 2$
$n_{1,2}(1{,}2) = 2$	$n_1(1{,}2) = 2$	$n_{1,2} = 3$
$n_{2,1}(2{,}3) = 2$	$n_1(3{,}4) = 3$	$n_{1,3} = 3$
$n_{1,3}(3{,}4) = 2$	$n_2(2{,}3) = 2$	$n_{2,1} = 2$
$n_{1,3}(0{,}1) = 1$	$n_3(1{,}2) = 1$	$n_{3,1} = 1$
$n_{3,3}(1{,}2) = 1$	$n_3(2{,}3) = 1$	$n_{3,3} = 1$
$n_{3,1}(2{,}3) = 1$		
$n_{1,2}(3{,}4) = 1$		

Table 5: Transition counts for sample data

Now, if we have observed a process with $\mathscr{s}$ states for T time periods and on n individuals there are $\mathscr{s} \times T \times n$ possible outcomes for the experiment. The likelihood function for all these outcomes is as follows:

$$L(P|\boldsymbol{X}) = \prod \left[p_{X_0,X_1}(0{,}1) p_{X_1,X_2}(1{,}2) \ldots p_{X_{T-1},X_T}(T-1,T) \right]^{n_{X_0,X_1,X_2,\ldots,X_T}} \qquad (36)$$

where the product is taken over all the $\mathscr{s} \times T \times n$ possible combinations of $X_0, X_1, X_2, \ldots, X_T$. Equation (36) can also be written as:

$$\begin{aligned} L(P|\boldsymbol{X}) &= \left(\prod \left[p_{X_0,X_1}(0{,}1) \right]^{n_{X_0,X_1,X_2,\ldots,X_T}} \right) \ldots \left(\prod \left[p_{X_{T-1},X_T}(T-1,T) \right]^{n_{X_0,X_1,X_2,\ldots,X_T}} \right) \\ &= \prod_{X_0,X_1} \left[p_{X_0^i,X_1^i} \right]^{n_{X_0,X_1}(0,1)} \ldots \prod_{X_{T-1},X_T} \left[p_{X_{T-1},X_T} \right]^{n_{X_{T-1},X_T}(T-1,T)} \\ &= \prod_{t=1}^{T} \prod_{g=1}^{\mathscr{s}} \prod_{j=1}^{\mathscr{s}} p_{gj}(t-1,t)^{n_{gj}(t-1,t)} . \end{aligned} \qquad (37)$$

We therefore only need to have the transition counts, $n_{gj}(t-1,t)$, to formulate the likelihood function. We therefore have that $n_{gj}(t-1,t); g,j = 1 \ldots \mathscr{s}; t = 1, \ldots, T$ forms a set of sufficient statistics for determining the ML estimators of a Markov chain.

The ML estimator for a stationary process will be discussed first and then generalised to the nonstationary case.

For a stationary process we have that $p_{gj}(t-1,t) = p_{gj}\ \forall\ t = 1, \dots, T$ and so the likelihood function in Equation (37) simplifies to:

$$\begin{aligned} L(P|\boldsymbol{X}) &= \prod_{t=1}^{T}\prod_{g=1}^{s}\prod_{j=1}^{s} {p_{gj}}^{n_{gj}(t-1,t)} \\ &= \prod_{g=1}^{s}\prod_{j=1}^{s} {p_{gj}}^{n_{gj}}. \end{aligned} \tag{38}$$

Now the likelihood in Equation (38) should be maximised by finding the values for p_{gj} that maximises the function under the restrictions that $\sum_{j=1}^{s} p_{gj} = 1$ and $0 \leq p_{gj} \leq 1$. For a fixed value of g in $p_{gj}; j = 1, \dots, s$, the transition probabilities are the same as that of a multinomial distribution that has been repeated $n_i = \sum_{j=1}^{s} n_{gj}$ times with observations $n_{g1}, n_{g2}, \dots, n_{gs}$. As shown in Appendix 1, the ML estimators of this multinomially distributed process are

$$\begin{aligned} \widehat{p_{gj}} &= \frac{n_{gj}}{n_g} \\ &= \frac{\sum_{t=1}^{T} n_{gj}(t-1,t)}{\sum_{t=1}^{T}\sum_{k=1}^{s} n_{gk}(t-1,t)} \end{aligned} \tag{39}$$

For a nonstationary process one could consider $n_{gj}(t-1,t)$ for a fixed g and t. Then the observations $n_{g1}(t-1,t), n_{g2}(t-1,t), \dots, n_{gs}(t-1,t)$ are the outcomes of a multinomial experiment that has been repeated $n_g(t-1,t)$ times with probabilities $p_{g1}(t-1,t)$, $p_{g2}(t-1,t), \dots, p_{gs}(t-1,t)$. So following similar arguments as above, we have that the ML estimator of a nonstationary Markov chain is

$$\widehat{p_{gj}}(t-1,t) = \frac{n_{gj}(t-1,t)}{\sum_{k=1}^{s} n_{gk}(t-1,t)}. \tag{40}$$

The ML estimators in Equations (39) and (40) make intuitive sense as the transition probability from state i to state j is estimated by the relative frequency of transitions from state i to state j. It should be noted that Equations (39) and (40) are valid for all Markov chains, irrespective of whether the chain is irreducible or not. Intuitively, it may seem that one would require a large sample size to implement Equation (40) but we show in Section 4.3.2 that the nonstationary estimator can work with relatively small sample sizes. This may however be specific to the process we implemented and it is adviced in practice that a simulation study be conducted on a process with a set of possible parameters for a process being studied if the sample size of a nonstationary process needs to be determined. It is expected that process with states that will be occupied for relatively short time periods should have bigger sample sizes than process in which all states are occupied for longer time periods.

Anderson and Goodman investigated the asymptotic distribution of the ML estimators and indicated that $\sqrt{n}\left(\widehat{p_{gj}} - p_{gj}\right)$ has a limiting normal distribution with mean 0 and variance $\frac{p_{gj}(1-p_{gj})}{\phi_g}$ where $\phi_g = \sum_{k=1}^{s}\sum_{t=1}^{T}\eta_k p_{kg}(t-1,t)$ and η_k represents the population proportion of processes for which the initial state is state k. For the s fixed, the variables $\sqrt{n\phi_g}\left(\widehat{p_{gj}} - p_{gj}\right)$ are asymptotically independent and have the same limiting distribution as other functions of the probabilities of a multinomial distribution with s samples, each of size $n\phi_g$. Anderson and Goodman uses the asymptotic distribution of the estimated transition probabilities to test hypothesis about the probabilities. These fall outside the scope of this dissertation and the interested reader is referred to Section 3 of the article by Anderson and Goodman[8] for more details.

3.3 Continuous time: Markov jump processes

The ML estimator of the transition rate matrix of a Markov jump process $\{X_t; 0 \leq t < \infty\}$ defined on (Ω, Σ, P) and based on n observations during the time interval $[0, T)$, will be discussed based on the article by Albert[4]. The ML estimator provided by Albert is only applicable to stationary processes and therefore this discussion will focus on stationary processes with some comments and suggestions being made about nonstationary processes. The method of Albert is based on a sample of Markov jump processes for which the state transitions are recorded exactly.

Similarly to the ML estimator of a Markov chain discussed in Section 3.2, we need to construct a likelihood function for the outcomes of a Markov jump process. The continuous time aspect of the Markov jump process makes the likelihood function more complicated than its discrete time counterpart since there is no upper limit on the number of transitions that can occur in a given time interval. We will therefore first construct a set of all possible sample functions for a Markov jump process that make k jumps in the time interval $[0, T)$.

The article by Albert derives the likelihood function by writing the probability of an outcome of a Markov jump process in terms of an integral over a set of possible values the process can obtain in the time interval $[0, T)$. The likelihood function is then derived using results from measure theory and finally maximised so that the ML estimators can be obtained. The derivation of the probability of an outcome of the process in terms of an integral will be discussed in detail. The steps involved in obtaining the likelihood function from this integral will not be discussed in detail since this involves extensive results from measure theory which fall beyond the scope of this dissertation.

Notation

Consider a Markov jump process with $\mathscr{s}$ states that is observed for the time interval $[0, T)$. Suppose that n outcomes of this experiment are observed and denote an outcome by $\omega \in \Omega$. Now let $\mathscr{t}_i(\omega)$ be the time of the i^{th} jump of the process with $\mathscr{t}_0(\omega) = 0$. We write $X_i(\omega)$ for the state of the process after the jump made at time $\mathscr{t}_i(\omega)$ so that $X_0(\omega)$ is the starting state of the process. Now, let $T_i(\omega) = \mathscr{t}_{i+1}(\omega) - \mathscr{t}_i(\omega)$ be the time spent in state $X_i(\omega)$. The realisation of the stochastic variable $T_i(\omega)$ will be written as $t_i(\omega)$. The stochastic variable $K(T)$ represents the total number of jumps observed in $[0, T)$ and its realisation is the largest integer k such that $\mathscr{t}_k(\omega) < T$. The observed process can therefore be written as follows:

$$\{X_t(\omega), 0 \leq t < T\} = \{(X_0(\omega), T_0(\omega)), \dots, (X_{k-1}(\omega), T_{k-1}(\omega)), X_k(\omega)\}. \qquad (41)$$

We therefore know the process jumped to state $X_k(\omega)$ at some time $\mathscr{t}_k(\omega) < T$ and is still in this state at time T but we do not observe the amount of time spent in this final observed state since the process is only observed until time T.

The ML estimator for the Markov jump process will be derived in terms of the transition rate matrix A. Only the off diagonal elements of A will be estimated since that the diagonal elements can be calculated using Equation (17). To simplify notation later on we define

$$\alpha_{ij}^* = \begin{cases} 0 & if\ j = i \\ \alpha_{ij} & if\ j \neq i \end{cases}.$$

We will write α_{ij} as $\alpha(i,j)$ to simplify notation and similarly, we write α^*_{ij} as $\alpha^*(i,j)$ and α_i as $\alpha(i)$.

<u>Set of possible sample functions and measures of the these sets</u>

Let the state space be represented by the set of integers $\mathcal{W}_0 = \{1,2,\dots,\mathscr{s}\}$. For an observed process with n jumps in $[0,T)$, the set of all possible sample functions can be represented by

$$\mathcal{W}_k = \left[\prod_{j=1}^{k}(\mathcal{W}_0 \otimes \mathbb{R})\right] \otimes \mathcal{W}_0 \tag{42}$$

where $\mathbb{R}$ denotes the real line and $\otimes$ a product of sets. The set of sample functions is then a vector consisting of the initial state, the states of the process at each jump and the length of time between jumps. Referring to the sample function discussed in Section 2.1, the part in brackets in Equation (42) is therefore represented by the first five solid horizontal lines in Figure 2 and the extra set with which the brackets in Equation (42) is multiplied by the last solid line. The set of possible sample functions can therefore be thought of as all the possible solid lines that can be filled into a figure similar to Figure 2 where it is understood that all the different lengths of the solid lines are included in the set of possible sample functions. Equation (41) is therefore an element of the set described by $\mathcal{W}_k$.

Now that we have defined the product set of the possible outcomes of the process, $\mathcal{W}_k$, we define

$$\sigma = \sum_{n=0}^{\infty}\left[\prod_{j=1}^{n}(C \times l)\right] \times C \tag{43}$$

to be a measure on the space of all possible sample functions, for all possible values of k. Here l is the Lebesgue measure on $\mathbb{R}$ and C a counting measure as defined in Definition 14 and Definition 10 respectively. The full measure-theoretic construction of σ can be found in the article by Albert[4]. The measure σ is therefore constructed to measure the state of the process and the time spent in the state for an observed Markov jump process with k transitions and σ is defined for all possible values of k.

The probability of an event as an integral

In the article by Albert[4] the following is shown:

$$\begin{aligned} P[K(T) = k, X_0 = x_0, T_0 \leq \tau_0, \dots, X_{k-1} = x_{k-1}, T_{k-1} \leq \tau_{k-1}, X_k = x_k\,] \\ = P[X_0 = x_0]e^{-\alpha(x_k)T} \\ \times \int_{S_k} \prod_{j=0}^{k-1} \alpha^*(x_j, x_{j+1}) e^{-[\alpha(x_j)-\alpha(x_k)]t_j}\, dt_j \;\; if\; k > 0 \end{aligned} \qquad (44)$$

where $S_k = \{(t_0, t_1, \dots, t_{k-1}) \colon \sum_{j=0}^{k-1} t_j < T \,\&\, 0 \leq t_j \leq \tau_j\}$.

Equation (44) is not written in terms of ω to simplify the notation. Therefore X_t is considered to be the same as $X_t(\omega)$. Now, if $k = 0$ Equation (44) becomes

$$\begin{aligned} P[K(T) = 0, X_0 = x_0\,] &= P[K(T) = 0|X_0 = x_0]P[X_0 = x_0] \\ &= P[X_t = x_0, K(T) = 0|X_0 = x_0]P[X_0 = x_0] \\ &= P[X_t = x_0 \;\forall\; t \in [0, T)|X_0 = x_0]P[X_0 = x_0] \\ &= e^{-\alpha(x_0)T}P[X_0 = x_0] \end{aligned} \qquad (45)$$

The last step in this derivation comes from the result in Equation (26).

Now, consider the case where $k > 0$. Since we are assuming the Markov property we have that

$$\begin{aligned} P[X_k = x_k|X_{k-1} = x_{k-1}, \dots, X_0 = x_0, T_{k-1} \leq \tau_{k-1}, \dots, T_0 \leq \tau_0] \\ = P[X_k = x_k|X_{k-1} = x_{k-1}, T_{k-1} \leq \tau_{k-1}], \end{aligned}$$

and from Equation (27) we have

$$P[X_k = x_k|X_{k-1} = x_{k-1}, T_{k-1} \leq \tau_{k-1}] = \frac{\alpha^*(x_{k-1}, x_k)}{\alpha(x_{k-1})}.$$

Therefore

$$P[X_k = x_k | X_{k-1} = x_{k-1}, \dots, X_0 = x_0, T_{k-1} \leq \tau_{k-1}, \dots, T_0 \leq \tau_0] \\ = \frac{\alpha^*(x_{k-1}, x_k)}{\alpha(x_{k-1})}. \tag{46}$$

Now, consider the following probability

$$P[T_k \leq \tau_k | X_k = x_k, \dots, X_0 = x_0, T_{k-1} \leq \tau_{k-1}, \dots, T_0 \leq \tau_0]. \tag{47}$$

From the Markov property and since we are considering a stationary process, Equation (47) becomes

$$P[T_k \leq \tau_k | X_k = x_k, \dots, X_0 = x_0, T_{k-1} \leq \tau_{k-1}, \dots, T_0 \leq \tau_0] = P[T_k \leq \tau_k | X_k = x_k].$$

The probability $P[T_k \leq \tau_k | X_k = x_k]$ is equivalent to the probability of leaving state X_k in the time interval $[T_{k-1}, T_{k-1} + \tau_k]$. From Equation (26) we have that

$$P[T_k \leq \tau_k | X_k = x_k] = 1 - e^{-\alpha(x_k)\tau_k}$$

since $e^{-\alpha(x_k)\tau_k}$ is the probability to remain in state X_k in the time interval $[T_{k-1}, T_{k-1} + \tau_k]$ given that $X_k = x_k$.

Therefore we have

$$P[T_k \leq \tau_k | X_k = x_k, \dots, X_0 = x_0, T_{k-1} \leq \tau_{k-1}, \dots, T_0 \leq \tau_0] = 1 - e^{-\alpha(x_k)\tau_k}. \tag{48}$$

Using the results from Equations (46) and (48) we have

$$\begin{aligned}
&P[X_k = x_k, T_k \leq \tau_k, X_{k-1} = x_{k-1}, T_{k-1} \leq \tau_{k-1}, \dots, X_0 = x_0, T_0 \leq \tau_0] \\
&= P[T_k \leq \tau_k | X_k = x_k, \dots, X_0 = x_0, T_{k-1} \leq \tau_{k-1}, \dots, T_0 \leq \tau_0] \\
&\qquad \times P[X_k = x_k, X_{k-1} = x_{k-1}, T_{k-1} \leq \tau_{k-1}, \dots, X_0 = x_0, T_0 \leq \tau_0] \\
&= \left(1 - e^{-\alpha(x_k)\tau_k}\right) P[X_k = x_k | X_{k-1} = x_{k-1}, T_{k-1} \leq \tau_{k-1}, \dots, X_0 = x_0, T_0 \leq \tau_0] \\
&\qquad \times P[X_{k-1} = x_{k-1}, T_{k-1} \leq \tau_{k-1}, \dots, X_0 = x_0, T_0 \leq \tau_0] \\
&= \left(1 - e^{-\alpha(x_k)\tau_k}\right) \frac{\alpha^*(x_{k-1}, x_k)}{\alpha(x_{k-1})} \\
&\qquad \times P[T_{k-1} \leq \tau_{k-1} | X_{k-1} = x_{k-1}, X_{k-2} = x_{k-2}, T_{k-2} \leq \tau_{k-2}, \dots, X_0 = x_0, T_0 \leq \tau_0] \\
&\qquad \times P[X_{k-1} = x_{k-1}, X_{k-2} = x_{k-2}, T_{k-2} \leq \tau_{k-2}, \dots, X_0 = x_0, T_0 \leq \tau_0] \\
&= \left(1 - e^{-\alpha(x_k)\tau_k}\right) \frac{\alpha^*(x_{k-1}, x_k)}{\alpha(x_{k-1})} \left(1 - e^{-\alpha(x_{k-1})\tau_{k-1}}\right) \\
&\qquad \times P[X_{k-1} = x_{k-1} | X_{k-2} = x_{k-2}, T_{k-2} \leq \tau_{k-2}, \dots, X_0 = x_0, T_0 \leq \tau_0] \\
&\qquad \times P[X_{k-2} = x_{k-2}, T_{k-2} \leq \tau_{k-2}, \dots, X_0 = x_0, T_0 \leq \tau_0]
\end{aligned}$$

$$= \left(1 - e^{-\alpha(x_k)\tau_k}\right)\frac{\alpha^*(x_{k-1}, x_k)}{\alpha(x_{k-1})}\left(1 - e^{-\alpha(x_{k-1})\tau_{k-1}}\right)\frac{\alpha^*(x_{k-2}, x_{k-1})}{\alpha(x_{k-2})}$$
$$\times P[X_{k-2} = x_{k-2}, T_{k-2} \leq \tau_{k-2}, \dots, X_0 = x_0, T_0 \leq \tau_0]$$

.

.

.

$$= \left(1 - e^{-\alpha(x_k)\tau_k}\right)\frac{\alpha^*(x_{k-1}, x_k)}{\alpha(x_{k-1})}\left(1 - e^{-\alpha(x_{k-1})\tau_{k-1}}\right)\frac{\alpha^*(x_{k-2}, x_{k-1})}{\alpha(x_{k-2})}\dots\left(1 - e^{-\alpha(x_1)\tau_1}\right)\frac{\alpha^*(x_1, x_2)}{\alpha(x_1)}$$
$$\times P[X_1 = x_1, X_0 = x_0, T_0 \leq \tau_0]$$
$$= \left(1 - e^{-\alpha(x_k)\tau_k}\right)\frac{\alpha^*(x_{k-1}, x_k)}{\alpha(x_{k-1})}\left(1 - e^{-\alpha(x_{k-1})\tau_{k-1}}\right)\frac{\alpha^*(x_{k-2}, x_{k-1})}{\alpha(x_{k-2})}\dots\left(1 - e^{-\alpha(x_1)\tau_1}\right)\frac{\alpha^*(x_1, x_2)}{\alpha(x_1)}$$
$$\times P[T_0 \leq \tau_0 | X_0 = x_0,\ X_1 = x_1]P[X_0 = x_0,\ X_1 = x_1]$$
$$= \left(1 - e^{-\alpha(x_k)\tau_k}\right)\frac{\alpha^*(x_{k-1}, x_k)}{\alpha(x_{k-1})}\left(1 - e^{-\alpha(x_{k-1})\tau_{k-1}}\right)\frac{\alpha^*(x_{k-2}, x_{k-1})}{\alpha(x_{k-2})}\dots\left(1 - e^{-\alpha(x_1)\tau_1}\right)\frac{\alpha^*(x_1, x_2)}{\alpha(x_1)}$$
$$\times\ \left(1 - e^{-\alpha(x_0)\tau_0}\right) P[\, X_1 = x_1 | X_0 = x_0]P[X_0 = x_0]$$
$$= \left(1 - e^{-\alpha(x_k)\tau_k}\right)\frac{\alpha^*(x_{k-1}, x_k)}{\alpha(x_{k-1})}\left(1 - e^{-\alpha(x_{k-1})\tau_{k-1}}\right)\frac{\alpha^*(x_{k-2}, x_{k-1})}{\alpha(x_{k-2})}\dots\left(1 - e^{-\alpha(x_1)\tau_1}\right)\frac{\alpha^*(x_1, x_2)}{\alpha(x_1)}$$
$$\times \left(1 - e^{-\alpha(x_0)\tau_0}\right)\ \frac{\alpha^*(x_0, x_1)}{\alpha(x_0)} P[Z_0 = x_0]$$
$$= \left(1 - e^{-\alpha(x_k)\tau_k}\right)\prod_{j=0}^{k-1}\left\{\frac{\alpha^*\left(x_j, x_{j+1}\right)}{\alpha\left(x_j\right)}\left(1 - e^{-\alpha\left(x_j\right)\tau_j}\right)\right\} P[X_0 = x_0].$$

We therefore have that

$$\begin{aligned} &P[X_k = x_k, T_k \leq \tau_k, X_{k-1} = x_{k-1}, T_{k-1} \leq \tau_{k-1}, \dots, X_0 = x_0, T_0 \leq \tau_0] \\ &\qquad = \left(1 - e^{-\alpha(x_k)\tau_k}\right)\prod_{j=0}^{k-1}\left\{\frac{\alpha^*\left(x_j, x_{j+1}\right)}{\alpha\left(x_j\right)}\left(1 - e^{-\alpha\left(x_j\right)\tau_j}\right)\right\} P[X_0 = x_0]. \end{aligned} \tag{49}$$

Now we would like to write Equation (49) as a integral over all the values of $t_j \in \left[0, \tau_j\right]\ \forall j = 0, \dots, k$ and then integrate over all values of τ_k so that a probability equation can be obtained that does not contain a condition for T_k. To do that we notice that

$$\begin{aligned}\frac{\partial^{k+1}}{\partial t_0 \dots \partial t_k} \prod_{j=0}^{k-1} \left\{ \frac{\alpha^*(x_j, x_{j+1})}{\alpha(x_j)} \left(1 - e^{-\alpha(x_j)t_j}\right) \right\} \left(1 - e^{-\alpha(x_k)t_k}\right) \\ = \left[\prod_{j=0}^{k-1} \alpha^*(x_j, x_{j+1}) e^{-\alpha(x_j)t_j} \right] \alpha(x_k) e^{-\alpha(x_k)t_k}.\end{aligned} \tag{50}$$

Equation (50) is therefore the antiderivative of

$$\prod_{j=0}^{k-1} \left\{ \frac{\alpha^*(x_j, x_{j+1})}{\alpha(x_j)} \left(1 - e^{-\alpha(x_j)t_j}\right) \right\} \left(1 - e^{-\alpha(x_k)t_k}\right)$$

with respect to t_j and so Equation (49) can be represented as a multi-integral with respect to $t_0, \dots, t_k$ of

$$\left[\prod_{j=0}^{k-1} \alpha^*(x_j, x_{j+1}) e^{-\alpha(x_j)t_j} \right] \alpha(x_k) e^{-\alpha(x_k)t_k} P[X_0 = x_0].$$

We know that T_k will be in the interval $[T - \sum_{j=0}^{k-1} t_j, \infty)$. Therefore if we integrate over $[T - \sum_{j=0}^{k-1} t_j, \infty)$, Equation (49) does not contain a T_k term anymore and becomes

$$P[X_k = x_k, T_{k-1} \le \tau_{k-1}, X_{k-1} = x_{k-1}, \dots, X_0 = x_0, T_0 \le \tau_0]$$

$$= \int_0^{\tau_0} \dots \int_0^{\tau_{k-1}} \int_{T - \sum_{j=0}^{k-1} t_j}^{\infty} P[X_0 = z_0] \alpha(x_k) e^{-\alpha(x_k)t_k} \prod_{j=0}^{k-1} \alpha^*(x_j, x_{j+1}) e^{-\alpha(x_j)t_j}\, dt_k \dots dt_0. \tag{51}$$

Now, perform the innermost integration of Equation (51) by integrating out t_k:

$$\begin{aligned}&P[X_k = x_k, X_{k-1} = x_{k-1}, T_{k-1} \le \tau_{k-1}, \dots, X_0 = x_0, T_0 \le \tau_0] \\ &= \int_0^{\tau_0} \dots \int_0^{\tau_{k-1}} \left\{ \int_{T - \sum_{j=0}^{k-1} t_j}^{\infty} \alpha(z_k) e^{-\alpha(x_k)\tau_k} dt_k \right\} P[X_0 = x_0] \prod_{j=0}^{k-1} \alpha^*(x_j, x_{j+1}) e^{-\alpha(x_j)t_j}\, dt_{k-1} \dots dt_0 \\ &= \int_0^{\tau_0} \dots \int_0^{\tau_{k-1}} \left\{ -e^{-\alpha(x_k)t_k} \Big|_{T - \sum_{j=0}^{k-1} t_j}^{\infty} \right\} P[X_0 = x_0] \prod_{j=0}^{k-1} \alpha^*(x_j, x_{j+1}) e^{-\alpha(x_j)t_j}\, dt_{k-1} \dots dt_0\end{aligned}$$

$$= \int_0^{\tau_0} \dots \int_0^{\tau_{k-1}} \left\{-e^{-\alpha(x_k)\infty} + e^{-\alpha(x_k)\left(T-\sum_{j=0}^{k-1} t_j\right)}\right\}$$

$$\times P[X_0 = x_0] \prod_{j=0}^{k-1} \alpha^*(x_j, x_{j+1}) e^{-\alpha(x_j)t_j}\, dt_{k-1} \dots dt_0 \qquad (52)$$

$$= \int_0^{\tau_0} \dots \int_0^{\tau_{k-1}} \left\{e^{-\alpha(x_k)T} e^{\alpha(x_k)\sum_{j=0}^{k-1} t_j}\right\} P[X_0 = x_0] \prod_{j=0}^{k-1} \alpha^*(x_j, x_{j+1}) e^{-\alpha(x_j)t_j}\, dt_{k-1} \dots dt_0$$

$$= P[X_0 = x_0] \int_0^{\tau_0} \dots \int_0^{\tau_{k-1}} e^{-\alpha(x_k)T} \prod_{j=0}^{k-1} \alpha^*(x_j, x_{j+1}) e^{-[\alpha(x_j)-\alpha(x_k)]t_j}\, dt_{k-1} \dots dt_0.$$

If we specify the area of integration in Equation (52) to be over the set S_k, which is defined in Equation (44), we get

$$P[K(T) = k, X_k = x_k, X_{k-1} = x_{k-1}, T_{k-1} \leq \tau_{k-1}, \dots, X_0 = x_0, T_0 \leq \tau_0]$$

$$= P[X_0 = x_0] \int_{S_k} e^{-\alpha(x_k)T} \prod_{j=0}^{k-1} \alpha^*(x_j, x_{j+1}) e^{-[\alpha(x_j)-\alpha(x_k)]t_j}\, dt_{k-1} \dots dt_0$$

since the set S_k implies that $K(T) = k$ by specifying $\sum_{j=0}^{k-1} t_j < T$. This integral is understood to be a multi-integral.

To summarise, we have shown that

$$P[K(T) = k, X_0 = x_0, T_0 \leq \tau_0, \dots, X_{k-1} = x_{k-1}, T_{k-1} \leq \tau_{k-1}, X_k = x_k\ ]$$

$$= \begin{cases} e^{-\alpha(x_0)T} P[X_0 = x_0] & if\ k = 0 \\ \int_{S_k} P[X_0 = x_0] e^{-\alpha(x_k)T} \prod_{j=0}^{k-1} \alpha^*(x_j, x_{j+1}) e^{-[\alpha(x_j)-\alpha(x_k)]t_j}\, dt_{k-1} \dots dt_0 & if\ k > 0 \end{cases} \qquad (53)$$

which proves Equation (44).

Albert shows that for an event B of the Markov jump process we have the following probability distribution function:

$$P[B] = \int_B f_A(v) d\sigma(v)$$

where

$$f_{\boldsymbol{A}}(v) = \begin{cases} e^{-\alpha(x_0)T}P[X_0 = x_0] & if\ v = (x_0) \\ P[X_0 = x_0]e^{-\alpha(x_k)T}\prod_{j=0}^{n-1}\alpha^*(x_j, x_{j+1})e^{-[\alpha(x_j)-\alpha(x_k)]t_j} & if\ v = \big((x_0, t_0), \dots, (x_{k-1}, t_{k-1}), x_k\big) \end{cases}$$

The maximum likelihood estimators

The likelihood function of n realisations, $v_1, v_2, \dots, v_n$, of the Markov jump process $\{X_t(\omega), t \geq 0\}$ can be written as

$$L_{\boldsymbol{A}}^{(n)} = \prod_{h=1}^{n} f_{\boldsymbol{A}}(v_h).$$

For n observations, let $K_T^{(n)}(i,j)$ be the number of transitions from state i to state j and $\mathcal{A}_T^{(n)}(i)$ the total amount of time that the process is in state i. Taking the natural logarithm of $L_{\boldsymbol{A}}^{(n)}$ we get

$$\begin{aligned} \ln\,(L_{\boldsymbol{A}}^{(n)}) &= nln\big(P[X_0 = x_0^h]\big) - \sum_{h=1}^{n}\alpha(x_k^h)T \\ &+ \sum_{h=1}^{n}\sum_{j=1}^{k-1} ln\left(\alpha^*(x_j^h, x_{j+1}^h)\right) - \sum_{h=1}^{n}\sum_{j=1}^{k-1}[\alpha(x_j^h) - \alpha(x_k^h)]t_j^h. \end{aligned} \tag{54}$$

The first term in Equation (54) does not depend on **A** and will therefore be denoted by $\mathrm{C_n}$ since it does not affect the likelihood in terms of **A**.

The terms in the summation of $\sum_{h=1}^{n}\sum_{j=1}^{k-1} ln\left(\alpha^*(x_j^h, x_{j+1}^h)\right)$ will only be nonzero over the values of i, j for which a jump is observed form state X_i to X_j such that $X_i \neq X_j \in (1, \dots \mathcal{s})$. Therefore $\sum_{h=1}^{n}\sum_{j=1}^{k-1} ln\left(\alpha^*(x_j^h, x_{j+1}^h)\right)$ can be written as $\sum_{i=1}^{\mathcal{s}}\sum_{\substack{j=1 \\ i\neq j}}^{\mathcal{s}} K_T^{(n)}(i,j)ln(\alpha_{ij})$.

The last term of Equation (54) can be written as

$$-\sum_{h=1}^{n}\sum_{j=1}^{k-1}[\alpha(x_j^h) - \alpha(x_k^h)]t_j^h = -\sum_{h=1}^{n}\sum_{j=1}^{k-1}\alpha(x_j^h)t_j^h + \sum_{h=1}^{n}\sum_{j=1}^{k-1}\alpha(x_k^h)t_j^h. \tag{55}$$

The last term of Equation (55) depends on j only through t_j. Therefore the second term of Equation (54) and Equation (55) can be joined,

$$\begin{aligned} -\sum_{h=1}^{n} \alpha(x_k^h)T - \sum_{h=1}^{n}\sum_{j=1}^{k-1} \alpha(x_j^h)t_j^h + \sum_{h=1}^{n}\sum_{j=1}^{k-1} \alpha(x_k^h)t_j^h \\ = -\left[\sum_{h=1}^{n}\sum_{j=1}^{k-1} \alpha(x_j^h)t_j^h + \sum_{h=1}^{n} \alpha(x_k^h)\left(T - \sum_{j=1}^{k-1} t_j^h\right)\right]. \end{aligned} \tag{56}$$

Here, $T - \sum_{j=1}^{k-1} t_j$ is the amount of time observed for the last state visited for each observed process. Therefore Equation (56) can be written as

$$-\left[\sum_{h=1}^{n}\sum_{j=1}^{k-1} \alpha(x_j^h)t_j^h + \sum_{h=1}^{n} \alpha(x_k^h)\left(T - \sum_{j=1}^{k-1} t_j^h\right)\right] = -\sum_{i=1}^{\mathcal{S}} \mathcal{A}_T^{(n)}(i)\alpha_i.$$

The likelihood function in Equation (54) can therefore be written as

$$\ln\left(L_{\boldsymbol{A}}^{(n)}\right) = C_n + \sum_{i=1}^{\mathcal{S}}\sum_{\substack{j=1 \\ i\neq j}}^{\mathcal{S}} K_T^{(n)}(i,j) ln(\alpha_{ij}) - \sum_{i=1}^{\mathcal{S}} \mathcal{A}_T^{(n)}(i)\alpha_i \tag{57}$$

Albert states that the Halmos-Savage factorization theorem can be applied to Equation (57) to prove that the set $\left\{K_T^{(n)}(i,j), \mathcal{A}_T^{(n)}(i)\right\}_{i\neq j}$ is a sufficient statistic for $\boldsymbol{A}$. The interested reader is referred to [44] for more information on the Halmos-Savage factorization theorem.

Differentiating the likelihood function in Equation (57) with respect to α_{ij} we note that

$$\frac{d}{d\alpha_{ij}}(\alpha_i) = \frac{d}{d\alpha_{ij}}\left(\sum_{i\neq j} \alpha_{ij}\right) = 1\,, i \neq j.$$

Therefore, we have

$$\frac{d ln\left(L_{\boldsymbol{A}}^{(n)}\right)}{d\alpha_{ij}} = \frac{K_T^{(n)}(i,j)}{\hat{\alpha}_{ij}} - \mathcal{A}_T^{(n)}(i)\,, \mathrm{i} \neq \mathrm{j}. \tag{58}$$

Setting Equation (58) equal to zero and solving for $\hat{\alpha}_{ij}^{(n)}$ we get that for $i \neq j$

$$\hat{\alpha}_{ij}^{(n)} = \frac{K_T^{(n)}(i,j)}{\mathcal{A}_T^{(n)}(i)}\ , i \neq j. \tag{59}$$

$\hat{\alpha}_{ii}^{(n)}$ is then calculated, similarly to Equation (17), as follows:

$$\hat{\alpha}_{ii}^{(n)} = -\sum_{i \neq j} \hat{\alpha}_{ij}^{(n)}. \tag{60}$$

The article by Albert also provides results of the large sample properties of the ML estimators which is stated below without proof. The interested reader is advised to consult the article of Albert for more details of the derivation of these results.

The sample size of a stochastic process can increase by increasing the number of observed processes (n), or by increasing the time for which the process is observed (T). The ML estimator of a Markov jump process has the following four properties which comes directly from the paper by Albert[4]:

1. For a fixed value of T, $\lim_{k \to \infty} \hat{\alpha}_{ij}^{(n)} = \alpha_{ij}$ if the probability of state i being occupied at least once is greater than zero

2. For a fixed value of T, the set of random variables $\left\{n^{\frac{1}{2}}\left(\hat{\alpha}_{ij}^{(n)} - \alpha_{ij}\right)\right\}_{i \neq j}$ are independent and asymptotically normally distributed with zero mean and variance equal to $\alpha_{ij} / \int_0^T P[X_t = i]dt$ if every state has a positive probability of being occupied.

3. For n fixed, $\lim_{T \to \infty} \hat{\alpha}_{ij}^{(n)} = \alpha_{ij}$ if one of the eigenvalues of A is zero and if all the cofactors on the diagonal of $\boldsymbol{A}$ are positive. The $(i,j)^{\text{th}}$ cofactor of a square matrix is calculated as $(-1)^{(i+j)}C_{ij}$ where C_{ij} is the determinant of the matrix formed by removing the i^{th} row and j^{th} column of the matrix[65].

4. For n fixed, the joint distribution of the set $\left\{T^{\frac{1}{2}}\left(\hat{\alpha}^{(n)}(i,j) - \alpha(i,j)\right)\right\}_{i \neq j}$ is asymptotically independent and normally distributed with zero mean and variance equal to $\alpha(i,j)\rho / A^{(i,i)}$ where ρ is the product of all non zero eigenvalues of $\boldsymbol{A}$ and $A^{(i,i)}$ is the cofactor of $\boldsymbol{A}$.

3.4 Alternative derivation of likelihood function by Tuma et al

Tuma et al.[104] has an alternative method to derive the likelihood function which does not delve into the extensive measure theory used by Albert[4]. The approach of Tuma et al. is focussed on the idea of providing a likelihood function that can handle right censored data and that all state transitions are recorded exactly when they occur. A brief overview of the likelihood function derivation by Tuma is provided below.

Let $w_{mi} = \begin{cases} 1 \text{ if the } m^{\text{th}} \text{ transition of the } i^{\text{th}} \text{ subject is observed} \\ 0 \text{ if the } m^{\text{th}} \text{ transition of the } i^{\text{th}} \text{ subject is censored} \end{cases}.$

Consider for example the i^{th} subject observed in a sample of Markov jump process with three states. Now suppose for example that the process is in state 1 at time t_0, then moves to state 2 at time t_1, followed by a move back to state 1 at time t_2 and then the process stays in the this state until the study ends. The time in state 1 from time t_2 is therefore right censored. For the example described we have $\{w_{1i}, w_{2i}, w_{3i}\} = \{1,1,0\}$.

Now, let

$$v_{mji} = \begin{cases} 1 \text{ if the } m^{\text{th}} \text{ transition of the } i^{\text{th}} \text{ subject is observed and consists of a jump to state } j \\ 0 \quad \text{otherwise} \end{cases}$$

where v_{0ji} is understood to equal unity if the i^{th} observed subject starts the process in state$j$.

Tuma et al. begins their derivation of the likelihood function by considering first the information arising from the first jump observed for each subject. Assume n processes are observed. The likelihood contribution following the first jump is[2]

$$L = \prod_{i=1}^{n} \prod_{j=1}^{\mathscr{s}} \left\{ \left[G_j(t_1|t_0)\right]^{(1-w_{1i})v_{0ji}} \left[f_j(t_1|t_0)\right]^{w_{1i}v_{0ji}} \prod_{k=1}^{\mathscr{s}} \left[\frac{\alpha_{jk}(t_0)}{\alpha_j(t_0)}\right]^{w_{1i}v_{0ji}} \right\}. \qquad (61)$$

For the i^{th} subject, $\left[\mathrm{G_j}(t_1|t_0)\right]^{(1-w_{1i})v_{0ji}}$ represents the survival function in the first state and this term is only different from one if the first transition is not observed (i.e. censored) in which case all the other terms in Equation (61) are equal to one. This implies that the only

[2] It should be noted that the paper by Tuma et al.[103] is believed to contain a typing error since $\prod_{k=1}^{\mathscr{s}} \left[\frac{\alpha_{jk}(t_0)}{\alpha_j(t_0)}\right]^{w_{1i}v_{0ji}}$ is written as $\prod_{k=1}^{\mathscr{s}} \left[\frac{\alpha_{jk}(t_0)}{\alpha_j(t_0)}\right]^{-w_{1i}v_{0ji}}$

information added to the likelihood function if the first transition is censored is the probability that the process stays in state j at least up to time t_1.

If however the first transition is observed, $\left[\mathrm{G_j}(t_1|t_0)\right]^{(1-w_{1i})v_{0ji}}$ equals unity and does not add any information to the likelihood function. In this case $\left[\mathrm{f_j}(t_1|t_0)\right]^{w_{1i}v_{0ji}}$ is the probability that the process leaves state j at time t_1 after the process was in state j at time t_0. This probability is multiplied by $\prod_{k=1}^{\mathscr{s}}\left[\frac{\alpha_{jk}(t_0)}{\alpha_j(t_0)}\right]^{w_{1i}v_{0ji}}$ where $\frac{\alpha_{jk}(t_0)}{\alpha_j(t_0)}$ is the probability that the process transitions from state j to state $k\ \in(1,\dots,\mathscr{s})$.

Each subject similarly contributes in this manner to the likelihood function. Now, from Equation (31), we have $f_j(t_1|t_0)=\alpha_j(t_0)G_j(t_1|t_0)$. Equation (61) can therefore be written as follows:

$$\begin{aligned}L&=\prod_{i=1}^{n}\prod_{j=1}^{\mathscr{s}}\left\{\left[G_j(t_1|t_0)\right]^{(1-w_{1i})v_{0ji}}\left[f_j(t_1|t_0)\right]^{w_{1i}v_{0ji}}\prod_{k=1}^{\mathscr{s}}\left[\frac{\alpha_{jk}(t_0)}{\alpha_j(t_0)}\right]^{w_{1i}v_{0ji}}\right\}\\&=\prod_{i=1}^{n}\prod_{j=1}^{\mathscr{s}}\left\{\left[G_j(t_1|t_0)\right]^{(1-w_{1i})v_{0ji}}\alpha_j(t_0)^{w_{1i}v_{0ji}}G_j(t_1|t_0)^{w_{1i}v_{0ji}}\prod_{k=1}^{\mathscr{s}}\left[\frac{\alpha_{jk}(t_0)}{\alpha_j(t_0)}\right]^{w_{1i}v_{0ji}}\right\}\\&=\prod_{i=1}^{n}\prod_{j=1}^{\mathscr{s}}\left\{G_j(t_1|t_0)^{v_{0ji}}\prod_{k=1}^{\mathscr{s}}\alpha_{jk}(t_0)^{w_{1i}v_{0ji}}\right\}.\end{aligned}$$

The likelihood function of the second transition follows similarly and equals

$$\begin{aligned}L&=\prod_{i=1}^{n}\prod_{j=1}^{\mathscr{s}}\left\{\left[G_j(t_2|t_1)\right]^{(1-w_{2i})v_{1ji}}\left[f_j(t_2|t_1)\right]^{w_{2i}v_{1ji}}\prod_{k=1}^{\mathscr{s}}\left[\frac{\alpha_{jk}(t_1)}{\alpha_j(t_1)}\right]^{w_{2i}v_{1ji}}\right\}\\&=\prod_{i=1}^{n}\prod_{j=1}^{\mathscr{s}}\left\{G_j(t_2|t_1)^{v_{1ji}}\prod_{k=1}^{\mathscr{s}}\alpha_{jk}(t_1)^{w_{2i}v_{1ji}}\right\}.\end{aligned}$$

Continuing with this reasoning, the likelihood function of all m jumps for each subject equals

$$\begin{aligned}L=\prod_{i=1}^{n}\prod_{m=1}^{\infty}\prod_{j=1}^{\mathscr{s}}&\left\{\left[G_j(t_m|t_{m-1})\right]^{(1-w_{mi})v_{m-1,ji}}\left[f_j(t_m|t_{m-1})\right]^{w_{mi}v_{m-1,ji}}\right.\\&\left.\times\prod_{k=1}^{\mathscr{s}}\left[\frac{\alpha_{jk}(t_{m-1})}{\alpha_j(t_{m-1})}\right]^{w_{mi}v_{m-1,ji}}\right\}\\=\prod_{i=1}^{n}\prod_{m=1}^{\infty}\prod_{j=1}^{\mathscr{s}}\prod_{k=1}^{\mathscr{s}}&\left\{\left[G_j(t_m|t_{m-1})\right]^{v_{m-1,ji}}\alpha_{jk}(t_{m-1})^{w_{mi}v_{m-1,ji}}\right\}.\end{aligned}\tag{62}$$

If the transition rates are stationary we have that the time between transitions follows an exponential distribution, as shown in Section 2.2.2.1, and therefore $G_j(t_m|t_{m-1}) = e^{-(t_m-t_{m-1})\alpha_j}$ and so Equation (62) becomes

$$L = \prod_{i=1}^{n}\prod_{m=1}^{\infty}\prod_{j=1}^{\mathscr{s}}\prod_{k=1}^{\mathscr{s}}\left\{\left[e^{-(t_m-t_{m-1})\alpha_j}\right]^{v_{m-1,ji}}\alpha_{jk}(t_{m-1})^{w_{mi}v_{m-1,ji}}\right\}. \qquad (63)$$

Now, taking the natural logarithm of Equation (63) we get

$$ln(L) = \sum_{i=1}^{n}\sum_{m=1}^{\infty}\sum_{j=1}^{\mathscr{s}}\sum_{k=1}^{\mathscr{s}}\left\{-(t_m - t_{m-1})\alpha_j v_{m-1,ji} + w_{mi}v_{m-1,ji}ln\left(\alpha_{jk}(t_{m-1})\right)\right\}. \qquad (64)$$

The first term in Equation (64) will only be different from zero if the $(m-1)^{\text{th}}$ jump is to state j and subsequently $\sum_{i=1}^{n}\sum_{m=1}^{\infty}\sum_{j=1}^{\mathscr{s}}\sum_{k=1}^{\mathscr{s}}(t_m - t_{m-1})\alpha_j v_{m-1,ji}$ is the same as the amount of the time the process spends in state j multiplied by α_j and then the sum over all values of $j \in (1, \dots, \mathscr{s})$.

The last term in Equation (64) will only be different from zero if the m^{th} observed transition is from state j to state k. Therefore $\sum_{\mathrm{i}=1}^{\mathrm{n}}\sum_{\mathrm{m}=1}^{\infty}\sum_{\mathrm{j}=1}^{\mathscr{s}}\sum_{\mathrm{k}=1}^{\mathscr{s}} w_{mi}v_{m-1,ji}ln\left(\alpha_{jk}(t_{m-1})\right)$ is, for $j, k \in (1, \dots, \mathscr{s})$, equal to the number of times the process is observed to transition from state j to state k multiplied by $ln\left(\alpha_{jk}(t_{m-1})\right)$.

We therefore see that Equation (64) is equivalent to Equation (57). The way in which Tuma et al. therefore proposes to handle right-censored observations is the same as that of Albert[4]. Tuma et al. does not provide a closed form expression of the values of α_{jk} that maximise the likelihood function, but mentions that a FORTRAN computer program has been written that can maximise the likelihood function.

The likelihood function of Tuma et al. does however provide the flexibility of specifying any distribution for the time spent in the j^{th} state which makes it possible to work with a nonstationary process.

Another advantage of the likelihood function proposed by Tuma et al. is that they mention that the transition rate can be estimated in terms of covariates, i.e. by specifying:

$$\alpha_{jk} = e^{\boldsymbol{\theta}_{jk}{}'\boldsymbol{z}}$$

where $\boldsymbol{\theta}_{jk}$ is a vector of parameters associated with characteristics, $\boldsymbol{z}$, of the units being studied. For patients, these characteristics can, amongst others, be smoking status and blood pressure. This flexibility in the transition rate can easily be added to the likelihood function of Albert by writing Equation (57) as

$$\ln\left(L_{\boldsymbol{A}}^{(n)}\right) = C_n + \sum_{h=1}^{n}\left(\sum_{i=1}^{s}\sum_{\substack{j=1\\ i\neq j}}^{s} K_T^{(h)}(i,j)\boldsymbol{\theta}_{ij}{}'\boldsymbol{z}_h - \sum_{i=1}^{s}\mathcal{A}_T^{(h)}(i)\alpha_h{}'(i)\right) \tag{65}$$

where $K_T^{(h)}(i,j)$ is the total number of transitions from state i to state j for the h^{th} process, $\mathcal{A}_T^{(h)}(i)$ is the total amount of time the h^{th} process spent in state i, $\alpha_h{}'(i) = \sum_{k\neq i} e^{\boldsymbol{\theta}_{ik}{}'\boldsymbol{z}_h}$ and $\boldsymbol{z}_h$ is the vector of covariates for the h^{th} process.

3.5 The method of Kalbfleisch and Lawless for irregularly observed data

Kalbfleisch and Lawless[53] define a likelihood function for a Markov jump process for which all processes in a sample are observed at the same irregular time points. Their method considers all the observed time points in a sample and that transitions can occur at the time of observation or during the start and end of an observation interval. The likelihood of observing all the transitions in the sample is then calculated similarly as in Equation (37). If the process is observed at times $t_0, t_1, \dots, t_m$, the likelihood is

$$L = \prod_{l=1}^{m}\prod_{i=1}^{s}\prod_{j=1}^{s} p_{ij}(t_{l-1}, t_l)^{n_{ij}(t_{l-1},t_l)} \tag{66}$$

where $n_{ij}(t_{l-1}, t_l)$ is the number of transitions from state i to j in the time interval $[t_{l-1}, t_l]$ as used in Section 3.2. Taking the natural logarithm of Equation (66), we get the log likelihood function:

$$ln(L) = \sum_{l=1}^{m}\sum_{i=1}^{s}\sum_{j=1}^{s} n_{ij}(t_{l-1}, t_l)ln\left[p_{ij}(t_{l-1}, t_l)\right]. \tag{67}$$

The approach by Kalbfleisch and Lawless considers all non-zero entries of the transition intensity matrix to be entries of a vector $\boldsymbol{\theta}$. For example, for the three state model with one absorbing state discussed in Section 4.2, $\boldsymbol{\theta} = (\lambda_1, \mu_1, \lambda_2, \mu_2)$ if we let

$$\boldsymbol{A} = \begin{bmatrix} -\lambda_1 - \mu_1 & \lambda_1 & \mu_1 \\ \lambda_2 & -\lambda_2 - \mu_2 & \mu_2 \\ 0 & 0 & 0 \end{bmatrix}.$$

Equation (67) should now be maximised with respect to $\boldsymbol{\theta}$. Kalbfleisch and Lawless proposes the use of a Gauss-Newton method which requires the derivative of Equation (67) with respect to $\boldsymbol{\theta}$. Recalling Equation (25) we have

$$\boldsymbol{P}^{(t)} = \boldsymbol{B} diag\left(e^{d_1 t}, \dots, e^{d_{\mathscr{s}} t}\right)\boldsymbol{B}^{-1}$$

where $d_1, \dots, d_{\mathscr{s}}$ are the eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$ is a matrix with columns containing the eigenvectors of $\boldsymbol{A}$ in the same order as $d_1, \dots, d_{\mathscr{s}}$. Kalbfleisch and Lawless shows that if we calculate $\frac{\partial \boldsymbol{A}}{\partial \theta_u}$ and let $\boldsymbol{G}^{(u)} = \boldsymbol{B}\frac{\partial \boldsymbol{A}}{\partial \theta_u}\boldsymbol{B}^{-1}$ we get

$$\frac{\partial \boldsymbol{P}^{(t)}}{\partial \theta_u} = \boldsymbol{B} V_u \boldsymbol{B}^{-1} \qquad (68)$$

where $\boldsymbol{V}_u$ is a $\mathscr{s} \times \mathscr{s}$ matrix with $(i,j)^{\text{th}}$ entry

$$\begin{matrix} g_{ij}{}^{(u)} \dfrac{\left(e^{d_i t} - e^{d_j t}\right)}{d_i - d_j} & if\ i \neq j \\ g_{ii}{}^{(u)} t e^{d_i t} & if\ i = j \end{matrix}$$

and $g_{ij}{}^{(u)}$ is the $(i,j)^{\text{th}}$ entry of $\boldsymbol{G}^{(u)}$.

We therefore have an easy method for calculating the derivative of $\boldsymbol{P}^{(t)}$ if we differentiate $\boldsymbol{A}$ with respect to $\boldsymbol{\theta}$. Kalbfleisch and Lawless proposes the following optimisation algorithm to estimate $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{iter} = \boldsymbol{\theta}_{iter-1} + \boldsymbol{M}(\boldsymbol{\theta}_{iter-1})^{-1} S(\boldsymbol{\theta}_{iter-1})\ ; \quad iter = 0,1, \dots. \qquad (69)$$

where

- $\boldsymbol{\theta}_0$ is a set of initial values for the parameters to be estimated;
- $S(\boldsymbol{\theta})$ is the b dimensional vector with u^{th} entry

$$S_u(\boldsymbol{\theta}) = \frac{\partial \ln\,(L)}{\partial \theta_u} = \sum_{l=1}^{m} \sum_{i=1}^{\mathscr{s}} \sum_{j=1}^{\mathscr{s}} n_{ijl} \frac{\partial p_{ij}(t_l)/\partial \theta_u}{p_{ij}(t_l)}$$

where $t_l = \mathscr{t}_l - \mathscr{t}_{l-1}$ and Equation (68) can be used to calculate $\frac{\partial p_{ij}(t_l)}{\partial \theta_u}$; and

- $\boldsymbol{M}(\theta)$ is the $b \times b$ matrix with $(u,v)^{\text{th}}$ entry

$$M_{uv}(\boldsymbol{\theta}) = \sum_{l=1}^{m}\sum_{i=1}^{s}\sum_{j=1}^{s}\frac{n_i(t_{l-1},t_l)}{p_{ij}(t_l)}\frac{\partial p_{ij}(t_l)}{\partial\theta_u}\frac{\partial p_{ij}(t_l)}{\partial\theta_v}$$

where $n_i(t_{l-1},t_l) = \sum_{j=1}^{s} n_{ij}(t_{l-1},t_l)$ as used in Section 3.2. The algorithm is repeated until $\boldsymbol{\theta}_{iter}$ and $\boldsymbol{\theta}_{iter-1}$ are sufficiently close and thereafter the last value of $\boldsymbol{\theta}_{iter}$ becomes the estimate of $\boldsymbol{\theta}$, i.e. $\widehat{\boldsymbol{\theta}}$. The matrix $\boldsymbol{M}\left(\widehat{\boldsymbol{\theta}}\right)^{-1}$ is an estimate of the asymptotic covariance matrix of $\widehat{\boldsymbol{\theta}}$. Furthermore it is indicated that the mean sojourn time in state i is estimated as $-\alpha_{ii}\left(\widehat{\boldsymbol{\theta}}\right)^{-1}$ and the asymptotic variance of the sojourn time is estimated with

$$\alpha_{ii}(\boldsymbol{\theta})^{-4}\sum_{u=1}^{b}\sum_{v=1}^{b}\frac{\partial\alpha_{ii}(\boldsymbol{\theta})}{\partial\theta_u}\frac{\partial\alpha_{ii}(\boldsymbol{\theta})}{\partial\theta_v}M^{uv}(\boldsymbol{\theta})\Bigg|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}}$$

where $M^{uv}(\boldsymbol{\theta})$ is the $(u,v)^{\text{th}}$ entry of $\boldsymbol{M}(\boldsymbol{\theta})^{-1}$.

Kalbfleisch and Lawless mention that the proposed algorithm can be used in the case of right censored data with the understanding that $n_i(t_{l-1},t_l)$ is the number of subjects in state i at time t_{l-1} for whom the state occupied at time t_l is known. Unlike the methods of Anderson and Goodman[8], Albert[4] and Tuma et al.[104], the algorithm of Kalbfleisch and Lawless does therefore not use the information of the amount of time spent in a state if the time is censored.

Covariates are included into the model by specifying the transition intensities to be functions of a covariate vector $\boldsymbol{Z} = \left(Z_1,\dots,Z_p\right)$ in the following manner:

$$A(\boldsymbol{Z}) = \left(\alpha_{ij}(\boldsymbol{Z})\right)$$

with

$$\begin{aligned}\alpha_{ij}(\boldsymbol{Z}) &= e^{\boldsymbol{Z}'\boldsymbol{\beta}_{ij}} \qquad \mathit{if}\ i \neq j\\ \alpha_{ii}(\boldsymbol{Z}) &= -\sum_{i\neq j}\alpha_{ij}(\boldsymbol{Z})\ \mathit{if}\ i = j\end{aligned}$$

and where $\boldsymbol{\beta}_{ij}$ are the covariate effects associated with the transition from state i to state j.

Say r distinct sets of covariates are observed in a sample of n subjects. Then r different transition matrices can be formed:

$$A_h = A(\boldsymbol{Z}_h) = \left(\alpha_{ij}(\boldsymbol{Z}_h)\right);\quad h\,\epsilon\,(1,\dots,r)$$

If we let $n_{ij}^{(h)}(t_{l-1}, t_l)$ be the number of subjects with covariates $\boldsymbol{Z}_h$ that transitioned from state i to state j between t_{l-1} and t_l we can write the likelihood in Equation (67) as

$$ln(L) = \sum_{l=1}^{m} \sum_{i=1}^{s} \sum_{j=1}^{s} n_{ij}^{(h)}(t_{l-1}, t_l) \ln\left[p_{ij}(w_l, \boldsymbol{Z}_h)\right]. \qquad (70)$$

Equation (70) can be maximised using Equation (69) with $S(\boldsymbol{\theta}) = \sum_{h=1}^{r} S^{(h)}(\boldsymbol{\beta})$ and $\boldsymbol{M} = \sum_{h=1}^{r} \boldsymbol{M}^{(h)}(\boldsymbol{\beta})$. The algorithm described above will increase computationally as r increases. It is therefore advised that covariates be grouped.

Kalbfleisch and Lawless used a Pearson-type test for goodness-of-fit which will be discussed in Section 3.7.

3.6 The method of Kay and Jackson when processes are observed at different irregular time points

Kay[56] provided a method for ML estimators of a Markov jump process of which all processes in a sample are observed at possibly different irregular time points. Such data will cause interval censored data since the time of state transitions will be known to be in a certain interval but the exact time of transition will be unknown. If the state of the process is measured at two time points, the state of the process is not necessarily known between these observation times.

Kay suggested that the true "alive" state from which a disease model makes a transition to the death state is often unknown due to irregular observation times, but the time at which the transition to the death state occurs is often known with precision of up to one day. Kay used his methods to estimate the parameters of a process with three transient states indicating cancer stages and a death state.

Consider a stationary disease process with s states of which the s^{th} state is the $death$ state. If k transitions are observed for the h^{th} patient, $h \in (1, \ldots, n)$, but the last observed transition is not to the $death$ state thereby making the time to death censored and it is known that the process is still in the state $X_{k(h)}^{(h)}$ at the end of the observation period, the likelihood contribution of such an observation is

$$l^{(h)} = \prod_{i=0}^{k^{(h)}-1} p_{X_i^{(h)},X_{i+1}^{(h)}}\left(t_i^{(h)}\right) ,h = 1, \dots, n. \tag{71}$$

If, however the k^{th} transition of the h^{th} patient is to the death state but the state of the process immediately before dying is unknown, the contribution to the likelihood function is

$$l^{(h)} = \prod_{i=0}^{k^{(h)}-2} \left[p_{X_i^{(h)},X_{i+1}^{(h)}}\left(t_i^{(h)}\right)\right] \sum_{i \neq S} p_{X_{k^{(h)}-1}^{(h)},X_i^{(h)}} \left(t_{k^{(h)}-1}^{(h)} - 1\right) p_{X_i^{(h)},X_{\mathscr{s}}}(1) , \tag{72}$$

$$h = 1, \dots, n .$$

Kay states that in some instances it may be known that the process in the not in the $death$ state at time T but the exact state of the process (i.e. in which of the transient states the process is in) is unknown. In this case the likelihood contribution of such a observation is:

$$l^{(h)} = \prod_{i=0}^{k^{(h)}-1} \left[p_{X_i^{(h)},X_{i+1}^{(h)}}\left(t_i^{(h)}\right)\right] \sum_{j=1}^{\mathscr{s}-1} p_{X_{k^{(h)}-1}^{(h)},X_j} \left(T - t_{k^{(h)}}^{(h)}\right) ,h = 1, \dots, n.$$

The full likelihood is then created by taking the product of all the $l^{(h)}$ terms and Kay states that the likelihood can be maximised using Newton-type methods.

In the documentation of the R library *msm*, Jackson[49] alters Equation (72) and writes the likelihood contribution if the process enters the death state from an unknown "alive" state as

$$l^{(h)} = \prod_{i=0}^{k^{(h)}-2} \left[p_{X_i^{(h)},X_{i+1}^{(h)}}\left(t_i^{(h)}\right)\right] \sum_{i \neq \mathscr{s}} p_{X_{k^{(h)}-1}^{(h)},X_i^{(h)}} \left(t_{k^{(h)}}^{(h)}\right) \alpha_{X_i^{(h)},X_{\mathscr{s}}}. \tag{73}$$

Kay inherently argues that if the last observed state is $X_{k^{(h)}-1}^{(h)}$ and the process is known to move to the dead state at time $t_{k^{(h)}}^{(h)}$ the likelihood contribution consist of the probability to move from state $X_{k^{(h)}-1}^{(h)}$ to any another "alive" state in $t_{k^{(h)}-1}^{(h)} - 1$ time units multiplied by the probability to move from this state to the $death$ state on one time unit. This is because Kay considers the time of death to be recorded with precision up to one day. Jackson on the other hand considers the likelihood contribution to equal the probability of moving from state $X_{k^{(h)}-1}^{(h)}$ to some "alive" state $X_i^{(h)}, i \neq \mathscr{s}$ in $t_{k^{(h)}}^{(h)}$ time units and then instantaneously moving to the $death$ state. Otherwise, Jackson uses the same approach as Kay for dealing with censoring. Jackson provides flexibility for observation times that are recorded with exact precision and where the state of the process is always observed when a transition is made. The likelihood function for such an observed process will be as follows:

$$l^{(h)} = \prod_{i=0}^{k^{(h)}} p_{X_i^{(h)},X_i^{(h)}}\left(t_i^{(h)}\right)\alpha_{X_i^{(h)},X_{i+1}^{(h)}} . \tag{74}$$

The *msm* library has an option using Equation (74) to calculate the likelihood function if the exact transition times are observed by stating *exacttimes=TRUE*. Jackson states that richer methods like the Cox regression model introduced in Section 2.3 can be used if exact transition times are known.

3.7 Analysing the goodness-of-fit of the model

Jackson[49] mentions two methods that can be used to assess the goodness-of-fit of a model such as ours. The first method consists of a graphical comparison of the expected number of patients (from the n observed processes) in a state with the observations at various time points. To calculate the observed number of patients in a state at time τ^*, if τ^* falls between two observed time points, one can assume that the process is in the state observed prior to τ^*. The validity of this assumption will depend on how often the process is observed.

In the case of a stationary Markov chain Equations (5) and (7) can be used to calculate the expected number of patients in each state at time τ^* if the distribution vector of Equation (5) is set equal to the number of patients in each of the states at time t_0. Equation (6) can be used in a similar manner for nonstationary Markov chains. Equation (23) can be used to calculate the TPM matrix from a transition intensity matrix in the case of a stationary Markov jump process and then Equation (5) and can be used as described above to calculate the distribution vector at time τ^*. If the transition intensity matrix of a nonstationary Markov jump process can be categorised into a number of stationary matrices, the TPM of each of these intensity matrices can be calculated with Equation (23) and then Equation (6) can be used in a similar manner as described for the nonstationary Markov chain to calculate the expected number of patients in each state at time τ^*. The observed and estimated number of patients in each state can then be compared graphically for various values of τ^* with the use of the *plot.prevalence.msm* function in the *msm* package of R. If the observed and estimated state prevalence values are close it is an indication of a good fit.

The second method mentioned by Jackson uses the observed and expected number of processes in a Pearson-type goodness-of-fit test statistic which was discussed by Aguirre-Hernandez and Farewell[3]. Kalbfleisch and Lawless[53] also used a Pearson-type test for goodness-of-fit, and Aguirre-Hernandez and Farewell[3] added to the ideas of Kalbfleisch and Lawless. Kalbfleisch and Lawless calculated the observed and expected number of patients transitioning between states i and j for each time interval t_j based on the estimated process parameters. The expected number of transitions from state i to j in the time interval $[t_{h-1}, t_h)$ is calculated as

$$E_{ijh} = \hat{p}_{ij}(t_{h-1}, t_h) \sum_{j=1}^{s} n_{ij}(t_{h-1}, t_h)$$

and then the goodness-of-fit test statistic is calculated as

$$\mathcal{T}_{KL} = \sum_{i=1}^{s} \sum_{j=1}^{s} \sum_{h=1}^{m} \frac{\left(O_{ijh} - E_{ijh}\right)^2}{E_{ijh}}. \tag{75}$$

The subscript KL in $\mathcal{T}_{KL}$ of Equation (75) is used to indicate it is the test statistic introduced by Kalbfleisch and Lawless. O_{ijh} is the number of patients that transitioned between states i and j in the time interval $[t_{h-1}, t_h)$ in the sample. Kalbfleisch and Lawless states that the test statistic follows an asymptotic chi-square distribution and uses this to test for the significance of the goodness-of-fit.

Aguirre-Hernandez and Farewell consider groups of patients based on their characteristics (covariates) and calculates the observed and expected number of transitions for each group. The grouping of observed and expected transitions also consists of intervals on the time axis and the time spent in a state. The observed number of transitions between state i and j in one such group, say l, is then calculated as

$$O_{i,j,l} = \sum I\left[X_{t_{i+1}} = j, X_{t_i} = i\right] \tag{76}$$

where the summation is taken over all units in group l and over all values of time included in the grouping. Here $I[\cdot]$ is an indicator function. The expected number of transitions in group

l is calculated by determining the estimated probability to transition from state i to j for each patient in group l and then summing over all patients in group l. The expected number of transitions is

$$E_{i,j,l} = \sum_{l,[t_i t_{i+1})\in l} \hat{P}\left[X_{t_{i+1}} = j, X_{t_i} = i\right]. \qquad (77)$$

The condition of $[t_i t_{i+1}) \in l$ in the summation of Equation (77) indicates that only time intervals included in group l should be included in the summation.

The Pearson-type goodness-of-fit test statistic is then calculated with the following formula:

$$\mathcal{T}_{AF} = \sum_{l} \frac{\left(O_{i,j,l} - E_{i,j,l}\right)^2}{E_{i,j,l}}.$$

Aguirre-Hernandez and Farewell state that $\mathcal{T}_{AF}$ does not follow a chi-square distribution since the estimated transition probability can differ for each process. They therefore estimate the distribution of $\mathcal{T}_{AF}$ by forming bootstrap samples. Each bootstrap sample is formed by simulating n Markov processes based on the estimated parameters, $\hat{\boldsymbol{\theta}}$, and then estimating the process parameters again and calculating $O_{i,j,l}$ and $E_{i,j,l}$ based on the bootstrap sample and process parameters estimated from the bootstrap sample respectively. Chapter 4 will discuss how a Markov chain and jump process can be simulated from the process parameters. A $(1-\alpha)100\%$ confidence interval for the test statistic is then formed from the sampled distribution of the test statistic and the model is said to have a significantly good fit if the observed value of $\mathcal{T}_{AF}$ is contained in the confidence interval.

The Pearson-type goodness-of-fit test with bootstrap samples can be calculated with the *pearson.msm* function of the *msm* package by specifying the *boot=TRUE* option.

3.8 Discussion

We have discussed various ML methods for the estimation of the parameters of Markov processes and indicated the influence of the timing of observations on the likelihood function. The possible observations schemes of a Markov process are summarised below:

- **All state transitions are observed exactly**. The estimator of Albert[4] which is discussed in Section 3.3 can be used for such data if the effect of covariates on transition rates are not investigated. The estimator of Tuma et al.[104] which is discussed in Section 3.4 can be used if the effects of covariates are of interest.

- **The states of all processes are observed at the same equidistant time points**. The estimator of Anderson and Goodman[8] which is discussed in Section 3.2 can be used for such data.

- **The states of all processes are observed at the same, but not necessarily equidistant time points**. The estimator of Kalbfleisch and Lawless[53] which is discussed in Section 3.5 can be used for such data.

- **The states of all processes are observed at possibly different time points which are irregular**. The methods of Kay[56] and Jackson[49] which are discussed in Section 3.6 can be used for such data.

The methods of Albert[4] and Anderson and Goodman[8] provide closed form expressions for the ML estimators of the Markov process parameters whereas the estimators of Tuma et al.[104], Kalbfleisch and Lawless[53], Kay[56] and Jackson[49] are found by using iterative optimisation procedures to find the values of the parameters that maximise the likelihood function. Kalbfleisch and Lawless[53] provide an optimisation algorithm in their article and functions like the *nlm* function in R can be used to find the parameters that maximise the likelihood functions of Tuma et al.[104], Kay[56] and Jackson[49].

We have discussed one nonstationary ML estimator of a Markov process for which all processes are observed at the same equidistant time points. This is the nonstationary estimator of Anderson and Goodman[8] which was discussed in Section 3.2. Aalen et al[1] used a similar approach in estimating nonstationary transition probabilities, but indicated that the time points at which observations are made do not have to be equidistant. Aalen et al. indicates that the TPM of a nonstationary Markov chain for the interval $[s,t]$ should be estimated by splitting $[s,t]$ into a partition of time $\{t_m\}$ such that each observed transition in the interval $[s,t]$ is contained in a unique time interval. The estimated TPM for the interval $[s,t]$ then becomes:

$$\hat{P}^{(s,t)} = \prod_{s<t_m<t_{m+1}\leq t} \hat{P}^{(t_m,t_{m+1})}.$$

Tuma et al.[104] provided a likelihood function for which the transition rates of a Markov jump process can be estimated as a function of covariates. Tuma mentions that the function can be applied to include the effect of time on transition rates. Consider for example a three state Markov jump process of which the transition from state i to state j is written as $\alpha_{ij} = e^{\beta_0+\beta_1 t}$. The effect of time on transition rates can therefore be estimated by finding the ML estimates of β_0 and β_1 with the use of optimisation methods such as the *nlm* function in R. The type of function used to link the transition rate to time, $e^{\beta_0+\beta_1 t}$ in this example, will depend on the relationship between time and the transition rate and it is suggested that process parameters be estimated for various link functions with the outcome of each method being compared using the observed and expected state prevalence plots described in Section 3.7.

Another approach to handling nonstationary process in the approach of Tuma et al.[104] is by specifying a survival function other than the exponential for the time spent in the jth state in Equation (62).

In dealing with nonstationary processes, Kalbfleisch and Lawless[53] and Ocana-Riola[81] suggested that the transition intensity matrix be split into piecewise homogenous transition intensities which are stationary over certain time intervals. Ocana-Riola goes further to suggest a method whereby different homogenous transition intensity matrices are estimated between all time periods for which transitions were observed. If any two consecutive transition intensity matrices are the same, the one observation time is ignored and the homogenous transition intensity matrices are calculated again. This process continues until all consecutive transition intensity matrices are different. This method may however be impractical if the time period between transitions are small and many different time periods are observed of which most give different transition intensity matrices.

It is also possible that the transition rate may not depend on the time from the start of the process, but on the time spent in a state. Such models are called semi-Markov models and are described in Meira-Machado et al.[73] A discussion of methods that deals with this type of processes will not be provided since it falls outside the scope of this dissertation but software will be mentioned in the concluding chapter that can incorporate semi-Markov processes.

The summary of the methods that have been given so far suggests that the treatment of time in Markov processes, discrete or continuous, be determined by the data available. One may however wish to work with a Markov chain process even though a Markov jump process was estimated from the data and vice versa. Calculating the TPM from a transition intensity matrix has been described in Chapter 2 so it is possible to estimate the parameters of a Markov chain process even if the process has been observed in such a way that would suggest a Markov jump process be used to estimate the process parameters. If the data is however used to estimate a TPM, it is not always possible to construct the corresponding transition intensity matrix. This is known as the problem of embeddability and is discussed by, amongst others, Singer and Spilerman[99]. An example of a TPM that is not embeddable is provided in Equation (78) below. This example is used in the article of Singer and Spilerman[99].

$$P = \begin{bmatrix} 0.15 & 0.35 & 0.50 \\ 0.37 & 0.45 & 0.18 \\ 0.20 & 0.60 & 0.20 \end{bmatrix}. \tag{78}$$

If $s = 2$ the TPM, P, is embeddable if and only if $tr(P) > 1$. There is however not a general set of rules if $s > 2$ that can be used to determine of a TPM is definitely embeddable, but in Section 3.1 of the article by Singer and Spilerman[99] some rules are provided which be used to determine if a TPM cannot be embedded as a transition intensity matrix. These rules will not be discussed in detail here, since this falls outside the scope of this dissertation.

As mentioned by Kalbfleisch and Lawless[53], embeddability does not have an influence on the estimation of the transition intensity matrix nor does it give an indication of the goodness-of-fit of the estimates. The methods provided in Chapter 3 can be used to estimate the transition intensity matrix from equidistant observations, with the methods depending on the assumptions being made on when the transitions occurred relative to the observation time. This will be explored in Chapter 4 where a Markov jump process will be fitted to data observed at equidistant time points and then the TPM will be derived from the transition intensity matrix and compared to the TPM estimate of the Markov chain methods from Section 3.2.

When state transitions of a Markov jump processes are not observed exactly, the likelihood function contains transition probabilities based on the duration between observations. See for example the approach by Kay and Jackson discussed in Section 3.6. When optimising likelihood functions that contain transition probabilities in terms of transition intensities like in Equations (71), (72), (73) and (74), one can calculate the corresponding transition probability for every transition rate using Equations (24) or (25) if the process is stationary. Tuma et al.[104] gives the transition probabilities of a three state stationary Markov jump process with one absorbing state in terms of the transition rates and not in matrix notation. Consider for example a stationary Markov jump process with transition intensity matrix given by:

$$\boldsymbol{A} = \begin{bmatrix} \alpha_1 & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_2 & \alpha_{23} \\ 0 & 0 & 0 \end{bmatrix}.$$

The corresponding transition probability of a move from state j to state k in t time units is then given by:

$$\begin{aligned}
p_{jj}(t) &= \frac{1}{\delta_1 - \delta_2}\left[(\alpha_k + \delta_1)e^{\delta_1 t} - (\alpha_k + \delta_2)e^{\delta_2 t}\right] \\
p_{jk}(t) &= \frac{\alpha_{jk}}{\delta_1 - \delta_2}\left[e^{\delta_1 t} - e^{\delta_2 t}\right]; j \neq k \\
p_{j3}(t) &= 1 + \frac{1}{\delta_1 - \delta_2}\left[\left(\alpha_{j3} + \delta_2\right)e^{\delta_1 t} - \left(\alpha_{j3} + \delta_1\right)e^{\delta_2 t}\right]; j \neq 3 \\
p_{3j}(t) &= 0; j \neq 3 \\
p_{33}(t) &= 1
\end{aligned} \qquad (79)$$

with

$$\begin{aligned}
\delta_1 &= -\frac{1}{2}\left[\alpha_1 + \alpha_2 + \sqrt{(\alpha_1 - \alpha_2)^2 + 4\alpha_{12}\alpha_{21}}\right] \\
\delta_1 &= -\frac{1}{2}\left[\alpha_1 + \alpha_2 - \sqrt{(\alpha_1 - \alpha_2)^2 + 4\alpha_{12}\alpha_{21}}\right].
\end{aligned}$$

It is stressed by Tuma that Equation (79) is only valid for a three state Markov jump process of which the third state is an absorbing state. Kay[56] and Kalbfleisch[53] also used Equation (79) to write the transition probabilities of a three state system with one absorbing state in terms of the transition intensities. Kay notes that obtaining transition probabilities in a format similar to Equation (79) for a process with more than three states is nontrivial and suggests that Equation (25) be used in such cases.

When estimating the transition rates of a three state Markov jump process with one absorbing state using an iterative optimisation method one therefore has the option of obtaining transition intensities and forming a TPM in the likelihood, or using Equation (79) to calculate transition probabilities in the likelihood. These alternatives will be explored in Section 4.5 and compared in terms of accuracy and computation time.

The methods of Anderson and Goodman[8] use information on the number of transitions observed and right censoring does not influence their methods. When including time in the estimators however, the treatment of right censored time in a state becomes a matter to question. Albert[4], Tuma et al.[104], Kay[56] and Jackson[49] includes the time spent in the last state in the likelihood function even if the time is censored whereas Kalbfleisch and Lawless[53] do not. In another article of which Tuma was the lead author[103] it was stressed that the time spent in the last observed state should be included in the likelihood even if it is censored.

Markov processes will be simulated in Chapter 4 for a specific set of process parameters so that the parameters estimates can be compared to the true data generating parameters. The goodness-of-fit methods discussed in Section 3.7 will also be used to analyse the fit of the models and then comments will be made on the ability of the goodness-of-fit methods to assess the model fit.

Chapter 4: Implementation of methods

4.1 Introduction

The theory for estimating the parameters of Markov processes under various assumptions regarding the observations of the Markov process were described in Chapter 3. We will now simulate Markov processes and implement these methods to estimate the parameters under various assumptions of the observation scheme.

The parameters of processes for which the exact time of state transitions are recorded can be estimated with the use of the methods of Albert[4] and Tuma et al.[104], which were discussed in Sections 3.3 and 3.4. These methods will be implemented first by simulating a Markov jump process and estimating the transition intensities based on the simulated data. The ability of these methods to estimate covariate effects on transition intensities will also be investigated.

The next type of observation scheme that will be considered is that of processes for which all observations are made at equidistant time points and this can be described by Markov chains due to the discretization of time. We will simulate Markov chains and implement the methods of Anderson and Goodman which were discussed in Section 3.2.

Next we will consider an observation scheme where all processes are observed at the same irregular time points. For this we will simulate Markov jump processes that are observed at predefined time points and use the methods of Kalbfleisch and Lawless[53] to estimate the parameters of these processes.

The last observation scheme that will be investigated is the case where all processes are observed at possibly different irregular time points. For this we will simulate Markov jump processes and observe each process at different randomly chosen time points and then use the methods of Kay[53] and Jackson[49] to estimate the process parameters.

All estimated parameters will also be compared to that of the *msm* package of R. The true process parameters will always be known so it is straightforward to analyse the goodness-of-fit. The methods discussed in Section 3.7 will also be used to analyse the goodness-of-fit and the ability of these methods to analyse goodness-of-fit will be discussed.

The methodology used to simulate Markov processes is described next. Separate strategies were used for simulating Markov chains and jump processes.

4.1.1 Simulating a Markov chain

Given the TPM of a Markov chain and the current state of the process, the state of the process at the next time point can be simulated as follows:

1. If the current state of the process is state j, row j of the TPM vector, $(p_{j1}, p_{j2}, \dots, p_{j\mathscr{s}})$, provides the probabilities that the process will be in each of the states at the next time point.

2. Generate a random variable from a multinomial distribution with sample size one and probability vector $(p_{j1}, p_{j2}, \dots, p_{j\mathscr{s}})$. The value of the random multinomial variable indicates the state of the process at the start of the next time point with the understanding that the process can be in the same state as at the current time point.

The above algorithm can be repeated for as many time points and number of samples as required.

4.1.2 Simulating a stationary Markov jump process

As shown in Equation (26), if a stationary Markov jump process is currently in state i the time until the state of the process changes follows an exponential distribution with rate parameter $\alpha_i = \sum_{\substack{\mathrm{k} \in \mathcal{W}_0 \\ \mathrm{k} \neq \mathrm{i}}} \alpha_{\mathrm{ik}}$. The probability that this will result in a jump to state $j \neq i$ is $\frac{\alpha_{ij}}{\alpha_i}$.

For a stationary Markov jump process that just moved to state i, the time in state i can therefore be simulated by generating a random exponentially distributed variable with rate parameter α_i. The state of the process after the transition can be simulated as a multinomially distributed variable with sample size one and probabilities $\left\{\frac{\alpha_{ij}}{\alpha_i}; i \neq j\right\}$.

4.2 Exactly observed transitions

We begin by describing a Markov jump process for which the transitions are observed exactly and implement the method suggested by Albert and discussed in Section 3.3. The method of Tuma et al.[104] discussed in Section 3.4 was then used to include the effect of covariates on transition intensities.

4.2.1 Exactly observed transitions based on estimator by Albert

Example I

A Markov jump process with three states where all patients start the process in state 1 and transition intensity matrix given as

$$A = \begin{bmatrix} -0.35 & 0.15 & 0.2 \\ 0.2 & -0.4 & 0.2 \\ 0 & 0 & 0 \end{bmatrix} \tag{80}$$

was simulated . It was assumed that state transitions are observed with exact precision. This transition intensity matrix represents a process where states 1 and 2 communicate and movements from states 1 and 2 to state 3 are possible, with state 3 being an absorbing state. This can be thought of as being a $\{well, ill, dead\}$ disease model.

The process was run with $n = 1000$ and $T = 50$ where n refers to the number of processes sampled and T refers to the maximum observation time for each process. The transition intensity matrix was then estimated using Equations (59) and (60) and the estimated transition intensity matrix is

$$\hat{A} = \begin{bmatrix} -0.3389036 & 0.1420505 & 0.1968531 \\ 0.1860823 & -0.3989666 & 0.2128843 \\ 0 & 0 & 0 \end{bmatrix}. \tag{81}$$

It took the process on average 4.97 time units to move to the absorbing state and all the simulated processes were in the absorbing state at time $T = 50$. The process was run for the possible combinations of $n = 100{,}1000$ and $T = 5, 10$ and the results of the estimator as described by Albert is shown in Table 6. It should be noted that $\mathcal{A}_{\mathrm{T}}^{(\mathrm{k})}(i)$ in Equation (59) represents the amount of time that state i was occupied during the experiment by the n samples and it includes the observed amount of time spent in the last state even if the actual time in the state is censored. This was described in deriving Equation (55).

The *crudeinits.msm* function in the *msm* package also estimates the $(i,j)^{\text{th}}$ transition intensity by dividing the number of transitions from state i to j by the amount of time spent in state i. The estimates calculated from the simulated data were compared to that the of *crudeinits.msm* function and were verified as being the same. The estimated transition intensity matrices are compared with the true process parameters by calculating the sum of squared differences between estimated and actual parameters for all the elements of the transition intensity matrices. This quantity will be denoted as “SSD” and is shown in Table 6.

Scenario	Parameters	Estimated transition intensity matrix	SSD
1	$n = 1000$ $T = 10$	$\hat{A} = \begin{bmatrix} -0.3506 & 0.1473 & 0.2043 \\ 0.1966 & -0.4103 & 0.2137 \\ 0 & 0 & 0 \end{bmatrix}$	0.0003220456
2	$n = 1000$ $T = 5$	$\hat{A} = \begin{bmatrix} -0.3490 & 0.1488 & 0.2003 \\ 0.1535 & -0.3608 & 0.2073 \\ 0 & 0 & 0 \end{bmatrix}$	0.003752216
3	$n = 100$ $T = 10$	$\hat{A} = \begin{bmatrix} -0.3761 & 0.1436 & 0.2325 \\ 0.1297 & -0.3612 & 0.2315 \\ 0 & 0 & 0 \end{bmatrix}$	0.00923141
4	$n = 100$ $T = 5$	$\hat{A} = \begin{bmatrix} -0.3424 & 0.1484 & 0.1941 \\ 0.1883 & -0.4623 & 0.2740 \\ 0 & 0 & 0 \end{bmatrix}$	0.00958304

Table 6: Estimated transition intensity matrices

Table 6 shows that Equations (59) and (60) provide accurate estimates of the transition intensities even if the sample size and total observation time decrease to the parameters considered. The reduction in total observation time from 10 to 5 time units with $n = 1000$ produces a relatively big difference in SSD when compared to the case when $n = 100$.

The advantage of observing transitions exactly compared to panel data that are observed at equidistant time points was compared by producing a transition count matrix (TCM) that would have been observed if the data was only observed at the end of each time period. . The estimated TPM from the panel data ($TPM_{Discrete}$) is shown in Table 7. The estimated one step TPM for the transition intensity matrix in Equation (81) (TPM_{Exact}) is calculated using Equation (24) and also shown in Table 7. The true one step TPM based on Equation (80) (TPM_{True}) is also shown in Table 7.

$\widehat{TPM}_{Discrete} = \begin{bmatrix} 0.7138 & 0.1043 & 0.1819 \\ 0.1381 & 0.6559 & 0.2032 \\ 0 & 0 & 0 \end{bmatrix}$
$\widehat{TPM}_{Exact} = \begin{bmatrix} 0.7143 & 0.1012 & 0.1845 \\ 0.1351 & 0.6733 & 0.1916 \\ 0 & 0 & 0 \end{bmatrix}$
$TPM_{True} = \begin{bmatrix} 0.7151 & 0.1036 & 0.1813 \\ 0.1382 & 0.6806 & 0.1813 \\ 0 & 0 & 0 \end{bmatrix}$

Table 7: TPM matrices based on panel observations, exact observations and the true one step process TPM

Table 7 shows that the TPM estimated from exactly observed transitions are closer to the true one step TPM of the process when compared to what would be calculated from panel data.

Example II

The possible gain in accuracy of the estimated process parameters by observing transitions exactly compared to panel data was further investigated by considering a more complicated process. Therefore a process with four transient states and one absorbing state was considered. The transition intensity matrix used for this is shown in Equation (82). This was chosen such that patients have a small probability to enter state 4 and once they do they will move out of the state quickly. This was chosen to assess how many transitions to state 4 would be missed if the process is only observed at fixed time intervals. The process was run with a small sample size ($n = 100$) and with $T = 5$ to make sure some process will not be in the absorbing state by the time observation ends since we want to assess how well the process can predict state prevalence beyond observation time.

$$A = \begin{bmatrix} -1.13 & 0.2 & 0.8 & 0.03 & 0.1 \\ 0.8 & -1.11 & 0.2 & 0.01 & 0.1 \\ 0.4 & 0.9 & -1.42 & 0.02 & 0.1 \\ 1.3 & 0.8 & 0.5 & -2.68 & 0.08 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (82)$$

The estimated one step TPM from observing the process only at discrete time points ($TPM_{Discrete}$), the estimated one step TPM derived from the estimated transition intensity matrix (TPM_{Exact}) and the true step TPM (TPM_{True}) is shown in Table 8. The state prevalence calculated from these TPMs are shown in Figure 3.

$\widehat{TPM}_{Discrete} = \begin{bmatrix} 0.4178 & 0.1867 & 0.2756 & 0.0133 & 0.1067 \\ 0.3412 & 0.3529 & 0.2471 & 0.0118 & 0.0471 \\ 0.25 & 0.2778 & 0.4000 & 0 & 0.0833 \\ 0.5 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
$\widehat{TPM}_{Exact} = \begin{bmatrix} 0.4485 & 0.1709 & 0.2865 & 0.0076 & 0.0865 \\ 0.3290 & 0.3916 & 0.1893 & 0.0058 & 0.0844 \\ 0.2367 & 0.2604 & 0.4011 & 0.0074 & 0.0943 \\ 0.4586 & 0.1854 & 0.2008 & 0.1022 & 0.0530 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
$TPM_{True} = \begin{bmatrix} 0.4387 & 0.1921 & 0.2663 & 0.0078 & 0.0950 \\ 0.3119 & 0.4197 & 0.1677 & 0.0055 & 0.0950 \\ 0.2467 & 0.3072 & 0.3448 & 0.0062 & 0.0951 \\ 0.3550 & 0.2605 & 0.2222 & 0.0740 & 0.0886 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Table 8: TPM matrices based on panel observations, exact observations and the true one step process TPM

Table 8 indicates that $\widehat{TPM}_{Exact}$ is closer to $\widehat{TPM}_{True}$ than $\widehat{TPM}_{Discrete}$ with the SSD being 0.246284 and 0.360099 respectively. We also see that none of the transitions from state 3 to state 4 were observed by the panel data but it was observed in the exact transitions. Furthermore, only transitions from state 4 to states 1 and 4 were observed in the panel data. The lack of observing transitions from and to state 4 lead to wrong state prevalence in state 4 for the panel data as seen in Figure 3.

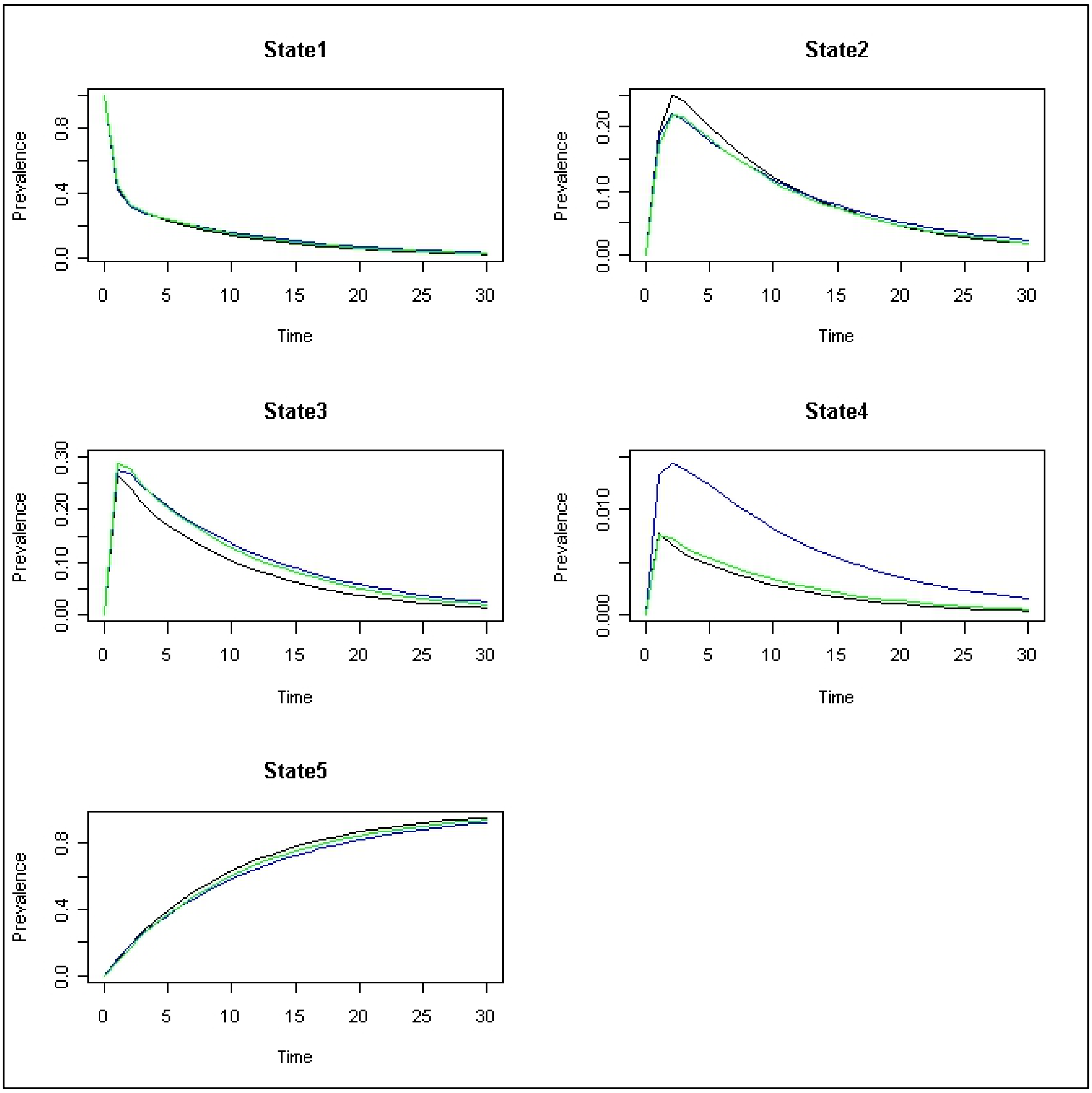

Figure 3: State prevalence. (blue line: based on $\widehat{TPM}_{Discrete}$, green line: $\widehat{TPM}_{Exact}$ and black line: TPM_{True}

4.2.2 The inclusion of covariate effects based on estimator by Tuma

The ML estimator of Tuma et al.[104] was discussed in Section 3.4. This method is based on observations that are made exactly when transitions occur and it was shown that the likelihood function of Tuma et al. is the same as that of Albert[4]. The estimator by Tuma et al. does not provide a closed form expression for the transition intensity estimates like that of Albert, but provides for easy addition of covariates to the likelihood function. The ML estimates can then be found by maximising the likelihood function using optimisation methods like that of the *nlm* function in R.

We start by considering a process without any covariate effects and whereby state transitions are observed exactly. This is similar to Example I and is done to assess the power of the *nlm* function to maximise the likelihood function. A three state Markov jump process with transition intensity matrix as in Equation (80) was simulated with $T = 50$ and $n = 1000$. Then the likelihood function in Equation (57) was maximised using the *nlm* function in R. This requires the calculation of the TCM, $K_T^{(n)}$, and the amount of time spent in every state $\mathcal{A}_T^{(n)}(i), i = 1, \dots \mathscr{s}$. The likelihood function was written in R such that the elements in the transition intensity matrix that had to estimated were elements of a vector $\boldsymbol{\theta} = (\lambda_1, \mu_1, \lambda_2, \mu_2)$ so that the transition intensity matrix can be written as

$$\boldsymbol{A} = \begin{bmatrix} -\lambda_1 - \mu_1 & \lambda_1 & \mu_1 \\ \lambda_2 & -\lambda_2 - \mu_2 & \mu_2 \\ 0 & 0 & 0 \end{bmatrix}. \tag{83}$$

The *nlm* function in R is an iterative method that requires a set of initial values for the parameters being estimated. The initial parameters were chosen as $\boldsymbol{\theta_0} = (0.2, 0.15, 0.15, 0.25)$ which are close to but not equal to the true process parameters, $\boldsymbol{\theta_{True}} = (0.15, 0.2, 0.2, 0.2)$. The parameters were estimated on the same data that was used to produce the estimated transition intensity matrix in Equation (81) and produced the following estimate for the transition intensity matrix

$$\hat{A} = \begin{bmatrix} -0.3389027 & 0.1420500 & 0.1968527 \\ 0.1860818 & -0.3989656 & 0.2128838 \\ 0 & 0 & 0 \end{bmatrix} \tag{84}$$

which is the same as Equation (81) up to four decimal points. We therefore see that using the *nlm* function in R to maximise the likelihood function in Equation (57) produces satisfactory results. The effect of the starting value on the estimates was assessed by using $\boldsymbol{\theta_0} =$

$(1,1,1,1)$ as initial vector and this produced results which were equal to Equation (84) up to six decimal points.

Example III

Next, the methodology proposed by Tuma to estimate the effect of covariates on transition rates was investigated. The same transition intensity matrix as Equation (80) was used to simulate processes, with the exception that the probability to move from state 1 to state 2 was assumed to be influenced by a binary variable z such that

$$\alpha_{12} = e^{\beta_{12,0}+\beta_{12,1}z}.$$

For this example, the binary variable z represents the gender of a patient and will therefore be denoted by Sex from here after with $Sex = 1$ representing a female patient and $Sex = 0$ a male patient. Then $e^{\beta_{12,0}+\beta_{12,1}}$ would represent the transition intensity from state 1 to state 2 of a female and $e^{\beta_{12,0}}$ that of a male patient. The transition intensity when the covariates are equal to zero, i.e. $e^{\beta_{12,0}}$ in this example, will be called the baseline transition intensity. The other non-zero off-diagonal transition intensities in Equation (80) were assumed to be independent of gender, or any other covariate effects, and were also written in the exponent. The simulated transition intensity matrix can therefore be written as

$$\boldsymbol{A} = \begin{bmatrix} -e^{\beta_{12,0}+\beta_{12,1}Sex} - e^{\beta_{13,0}} & e^{\beta_{12,0}+\beta_{12,1}Sex} & e^{\beta_{13,0}} \\ e^{\beta_{21,0}} & -e^{\beta_{21,0}} - e^{\beta_{23,0}} & e^{\beta_{23,0}} \\ 0 & 0 & 0 \end{bmatrix}$$

and the objective now becomes to estimate $\boldsymbol{\theta} = (\beta_{12,0}, \beta_{12,1}, \beta_{13,0}, \beta_{21,0}, \beta_{23,0})$. The sample was simulated with $n = 1000$, $T = 50$ and $\beta_{12,1} = 0.4$. The transition intensity from state 1 to state 2 of female patients is therefore $e^{\beta_{12,1}} = e^{0.4} = 1.491825$ times, or roughly 50%, greater than that of males. The gender of each patient was simulated from a Bernoulli variable with parameter $p = 0.7$ so that we obtain a sample with 70% females. The equivalent values of $\beta_{12,0}, \beta_{13,0}, \beta_{21,0}, \beta_{23,0}$ to obtain a similar transition intensity matrix as in Equation (80) were

$$\begin{gathered} \beta_{12,0} = ln(0.15) = -1.89712 \\ \beta_{13,0} = \beta_{21,0} = \beta_{23,0} = ln(0.2) = -1.60944 \end{gathered} \qquad (85)$$

so that the male patients were simulated from the same transition intensity matrix as Equation (80).

The likelihood function in Equation (65) was maximised with respect to the simulated values which required that a TCM for all 1000 simulated patients had to be calculated individually and also the state occupation times of each patient. This can be simplified by only calculating the TCM of all the males and females and also the total state occupation times for the males and females but this was programmed in R in a more general way so that continuously valued covariates could be added with ease.

The initial value for the *nlm* function was set to $\boldsymbol{\theta}'_0 = (ln(0.2), 0.7, \ln(0.15), ln(0.15), ln(0.2))$ so that the results could be compared to the example discussed above where the gender effect was not taken into account and the likelihood was maximised using the *nlm* function.

The likelihood function was optimised using the *nlm* function in R and the estimated and true values of the parameters are shown in Table 9. The parameters were also estimated with $\boldsymbol{\theta}''_0 = (ln(2), 1.5, ln(2), ln(2), ln(2))$ as starting value and the results of this is also shown in Table 9.

Parameter	True value	Estimated value with $\boldsymbol{\theta}'_0$ as starting value	Estimated value with $\boldsymbol{\theta}''_0$ as starting value
$\beta_{12,0}$	-1.89712	-2.0051925	-2.0051859
$\beta_{13,0}$	-1.60944	-1.6403000	-1.6402966
$\beta_{21,0}$	-1.60944	-1.6415876	-1.6415959
$\beta_{23,0}$	-1.60944	-1.6115572	-1.6115529
$\beta_{12,1}$	0.4	0.5337707	0.5337633

Table 9: Estimated and actual values for jump process with gender $(\boldsymbol{\beta}_{12,1})$ effect for different starting values for the *nlm* function

The estimated process parameters with $\boldsymbol{\theta}'_0$ as starting value produces the following estimated transition intensity matrix for the females

$$\hat{A} = \begin{bmatrix} -0.4235213 & 0.2295989 & 0.1939224 \\ 0.1936712 & -0.3932494 & 0.1995782 \\ 0 & 0 & 0 \end{bmatrix} \qquad (86)$$

and the following estimated transition intensity matrix for the males

$$\hat{A} = \begin{bmatrix} -0.3285562 & 0.1346338 & 0.1939224 \\ 0.1936712 & -0.3932494 & 0.1995782 \\ 0 & 0 & 0 \end{bmatrix}. \qquad (87)$$

Equation (87) is similar to Equation (84) indicating a good fit of the transition intensity matrix for the males. The estimated gender effect, $\hat{\beta}_{12,1} = 0.5337707$, is 30% higher than the true gender effect $\beta_{12,1} = 0.4$ and therefore overestimated.

We see that the estimates resulting from $\boldsymbol{\theta}_0'$ and $\boldsymbol{\theta}_0''$ as initial value in the *nlm* function were similar indicating the initial values do not have a big effect. Jackson[49] recommends that initial values of transitions intensities be set equal to the estimates of Equations (59) and (60). It is therefore recommended that the parameters of the transition intensities that are not functions of covariates be set to the natural logarithm of the estimates from Equations (59) and (60). When providing initial estimates for the binary variable one should investigate the effect of various "reasonable" parameter estimates, preferably based on prior knowledge of the process if possible. If we denote the initial value of the covariate effect by $\tilde{\beta}_{12,1}$, the proportion of females in the sample as $\overline{Sex}$, and the estimate of Equations (59) and (60) for the transition intensity from state 1 to state 2 with no covariate effect by $\hat{\alpha}_{12}$, the initial value for the baseline transition from state 1 to state 2 should be set equal to $ln(\hat{\alpha}_{12}) - \tilde{\beta}_{12,1}\overline{Sex}$. Various values of $\tilde{\beta}_{12,1}$ were considered as initial values in the *nlm* function with the rest of the initial values calculated as described above. It was observed that estimates similar to that of Table 9 were obtained when $-15 \leq \tilde{\beta}_{12,1} \leq 30$. $\tilde{\beta}_{12,1} = -15$ and $\tilde{\beta}_{12,1} = 30$ implies that the transition intensity from state 1 to state 2 of females will be respectively 3.059023×10^{-7} and 1.068647×10^{13} times that of males. If it is known from prior knowledge that the transition intensity of females cannot be less than 3.059023×10^{-7} or more than 1.068647×10^{13} times that of males it would make sense to use $-15 \leq \tilde{\beta}_{12,1} \leq 30$ when obtaining the ML estimates.

The *msm* package can estimate covariate effects by specifying *covariates* in the *msm* function. The msm package estimates the covariate effect for all transition intensities with a

95% confidence interval. The following estimated covariate effects were obtained with the *msm* package:

$$\begin{bmatrix} 0 & 0.5338\ (0.3585; 0.7090) & 0.0695\ (-0.0905; 0.2295) \\ -0.2366\ (-0.4767; 0.0035) & 0 & 0.1083\ (-0.1495; 0.3661) \\ 0 & 0 & 0 \end{bmatrix}. \quad (88)$$

We see that gender effect on the transition intensity from state 1 to state 2 estimated by the *msm* function is the same as that estimated by the *nlm* function. Furthermore we see that it is the only element of Equation (88) for which the 95% CI does not contain zero indicating that the effect is significant.

Example IV

Next, the addition of a continuous valued covariate was assessed by adding an age effect on the transition intensity from state 1 to state 2. Age here represents the age at study initiation and therefore remains constant for each patient throughout the study. One thousand Markov jump processes with $T = 50$ and the following transition intensity matrix were simulated,

$$\boldsymbol{A} = \begin{bmatrix} -e^{\beta_{12,0}+\beta_{12,1}Sex+\beta_{12,2}Age} - e^{\beta_{13,0}} & e^{\beta_{12,0}+\beta_{12,1}Sex+\beta_{12,2}Age} & e^{\beta_{13,0}} \\ e^{\beta_{21,0}} & -e^{\beta_{21,0}} - e^{\beta_{23,0}} & e^{\beta_{23,0}} \\ 0 & 0 & 0 \end{bmatrix}.$$

$\beta_{12,2}$ represents the effect of age on the transition rate and was simulated with $\beta_{12,2} = 0.05$. This means that the transition intensity from state 1 to state 2 would be $e^{0.05} = 1.051271$ times or roughly 5% more for every unit increase in age. Age represents a age vector and Sex is the gender vector with $Sex = 1$ being female and $Sex = 0$ being male. The process was again simulated with $\beta_{12,1} = 0.4$. Gender was simulated from a Bernoulli variable with probability parameter 0.7 and age was simulated from a $Unif(0,100)$ variable. The value of the other parameters for the simulation were chosen as follows:

$$\beta_{12,0} = ln(0.15) = -1.89712$$

$$\beta_{13,0} = \beta_{21,0} = \beta_{23,0} = ln(0.2) = -1.60944.$$

The transition intensity from state 1 to state 2 as a function of age and gender is shown in Figure 4.

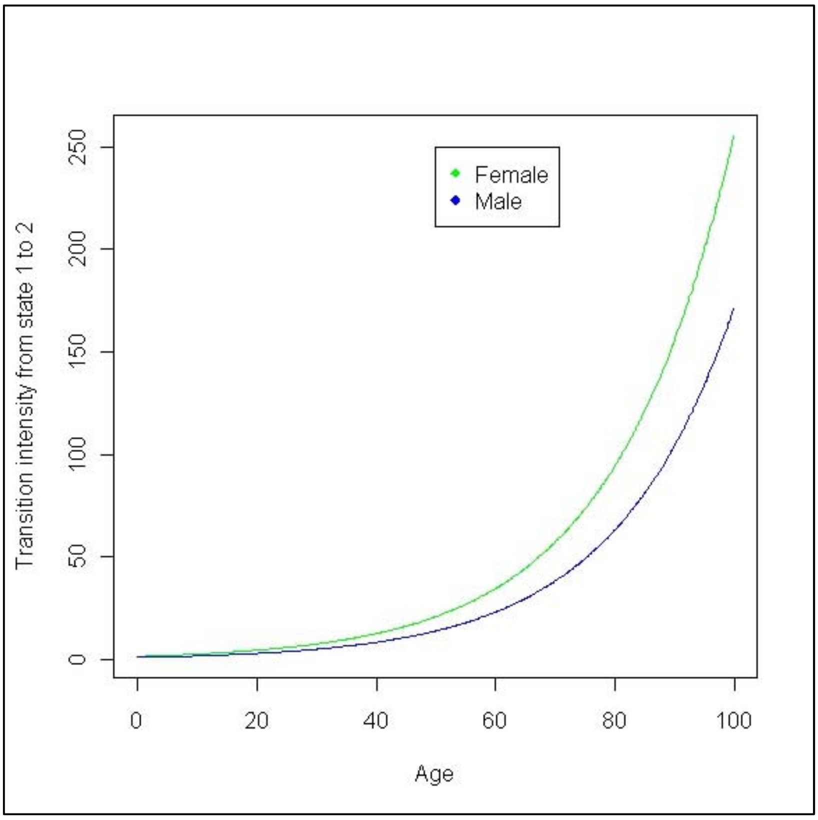

Figure 4: Transition intensity from state 1 to state 2 as a function of age and gender

The goal now is to find ML estimates for the vector $\boldsymbol{\theta} = (\beta_{12,0}, \beta_{12,1}, \beta_{12,2}, \beta_{13,0}, \beta_{21,0}, \beta_{23,0})$. The likelihood function in Equation (65) was maximised with the *nlm* function with $\boldsymbol{\theta}_{\mathbf{0}}' = (ln(0.2), 0.7, 0.1, ln(0.15), ln(0.15), ln(0.2))$ as initial value. The estimated and true values of the parameters are shown in Table 10. The effect of the starting value for the *nlm* function was assessed by using $\boldsymbol{\theta}_{\mathbf{0}}'' = (ln(2), 1.5, 1, ln(2), ln(2), ln(2))$ and these estimates are also shown in Table 10.

Parameter	True value	Estimated value with θ_0' as starting value	Estimated value with θ_0'' as starting value
$\beta_{12,0}$	-1.89712	-1.8287992	-0.2259975
$\beta_{13,0}$	-1.60944	-1.55452127	0.6931472
$\beta_{21,0}$	-1.60944	-1.5735262	0.6931472
$\beta_{23,0}$	-1.60944	-1.61257911	0.6931472
$\beta_{12,1}$	0.4	0.30857580	0.5808551
$\beta_{12,2}$	0.05	0.04998602	-89.9771531

Table 10: Estimated and actual values for jump process with gender $(\boldsymbol{\beta}_{12,1})$ and age $(\boldsymbol{\beta}_{12,2})$ effect with different starting values for the *nlm* function

Table 10 indicates that reasonable parameter estimates were obtained when $\boldsymbol{\theta}_0'$ was used as initial value in the *nlm* function. Similarly to Example III, we see that the choice of initial value in the *nlm* function can have a big influence on the parameter estimates. We have made suggestions in Example III for choosing initial values of the parameters corresponding to transition intensities that are not functions of covariates and also for effects associated with binary variables. When choosing the initial value for $\beta_{12,2}$ and $\beta_{12,1}$ in this example an approach was investigated in which the estimated gender effect estimated in Example III was used as initial value, i.e. $\tilde{\beta}_{12,1} = 0.5337752$, and various initial values for $\beta_{12,2}$ were considered. If we denote the initial estimate of the age effect by $\tilde{\beta}_{12,2}$ and the mean age in the sample by $\overline{Age}$, the corresponding initial estimate of $\beta_{12,0}$ would be $\tilde{\beta}_{12,0} = ln(\hat{\alpha}_{12}) - \tilde{\beta}_{12,1}\overline{Sex} - \tilde{\beta}_{12,2}\overline{Age}$ with $\hat{\alpha}_{12}$ having a similar interpretation as in Example III. We saw that similar estimates to Table 10 with $\boldsymbol{\theta}_0'$ as initial value were obtained when $\tilde{\beta}_{12,2}$ was within the interval $[-0.25, 0.35]$. If $\tilde{\beta}_{12,2} = -0.25$ the transition intensity of a person of age 100 would be $e^{-0.25*100} = 1.388794 \times 10^{-11}$ times than that of a new born. Similarly, if $\tilde{\beta}_{12,2} = 0.3$ the transition intensity of a person of age 100 would be $e^{0.3*100} = 1.586013 \times 10^{15}$ times than that of a new born. Practitioners should ask themselves if these estimates sound reasonable when using values $\tilde{\beta}_{12,2} = (-\infty, -0.25) \cup (0.35, \infty)$ as initial values.

The *msm* packages estimated the following effects of gender on the transition intensities

$$\begin{bmatrix} 0 & 0.3084\ (0.1113; 0.5056) & -0.05258\ (-0.6614; 0.5562) \\ -0.1206\ (-0.3993; 0.1582) & 0 & 0.1192\ (-0.1623; 0.4008) \\ 0 & 0 & 0 \end{bmatrix} \tag{89}$$

and the following effects of age on transition intensities,

$$\begin{bmatrix} 0 & 0.04999\ (0.04637; 0.05361) & -0.0015\ (-0.0106; 0.0077) \\ 0.0024\ (-0.0027; 0.0075) & 0 & -0.0035\ (-0.0087; 0.0017) \\ 0 & 0 & 0 \end{bmatrix}. \tag{90}$$

The brackets in Equation (89) and (90) represent a 95% CI for the covariate effects. Equation (89) and (90) indicates that the effects of gender and age are only significant in the transition intensities from state 1 to state 2. Furthermore, we see that the covariate effects estimated by the msm function are similar to that in Table 10 when $\boldsymbol{\theta}_0'$ was used as initial value in the *nlm* function.

4.3 Processes that are observed at equidistant time points

We showed in Section 3.2 that the parameters of processes that are observed at equidistant time points can be estimated by considering the processes to be Markov chains and then estimating the transition probabilities of such processes. The methodology discussed in Section 4.1.1 was used to simulate Markov chains. Equations (39) and (40) were then used to estimate the TPMs so that the ability of Equations (39) and (40) as estimators can be investigated to provide robust estimators. A stationary and nonstationary Markov chain were considered and are discussed separately.

4.3.1 Stationary Markov chain

Example V

A stationary Markov process with five states, of which the fifth state is absorbing, was considered for the estimation of a stationary process. The absorbing state resembles the death state of a disease model and the states are ranked by severity of disease with the first state representing the least severe form of the disease. The transition probabilities were chosen such that the probability to enter the absorbing state increases as disease severity increases to resemble a disease model whereby the risk of death increases as patients move to more severe disease states. The probability to move back to states of less severe disease decreases as the disease severity increases to simulate a disease where the chances of being cured decreases as the severity of disease increases. The TPM used to simulate the process is shown in Equation (91).

$$P = \begin{bmatrix} 0.7 & 0.1 & 0.05 & 0.05 & 0.1 \\ 0.13 & 0.2 & 0.35 & 0.2 & 0.12 \\ 0.1 & 0.05 & 0.1 & 0.5 & 0.25 \\ 0.05 & 0.05 & 0.1 & 0.4 & 0.4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (91)$$

Using Equation (9), we estimate that if the process starts in state 1 it will take 6.07 time units before entering the absorbing fifth state. First, $n = 1000$ Markov chains were sampled, each with $T = 50$ to ensure that all chains reach the absorbing state. All processes were started in state 1. A frequency count matrix (FCM) was calculated such that the $(i,j)^{\text{th}}$ element of the FCM represents $\sum_{t=1}^{T} n_{ij}(t-1,t)$. The *msm* package of R has a function *statetable.msm* which also calculates a FCM and we checked our FCM with that of the *statetable.msm* function to ensure that it was calculated correctly. Once the FCM is constructed it is straightforward to implement Equation (39) to estimate the TPM. The estimated TPM for this process is as follows

$$\hat{P} = \begin{bmatrix} 0.715 & 0.102 & 0.042 & 0.050 & 0.091 \\ 0.125 & 0.180 & 0.364 & 0.213 & 0.118 \\ 0.098 & 0.049 & 0.094 & 0.455 & 0.305 \\ 0.045 & 0.038 & 0.105 & 0.393 & 0.419 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (92)$$

If we compare Equations (91) and (92), it seems like the ML estimator is a robust measure for estimating the TPM of a stationary Markov chain. As a measure for the goodness-of-fit of the estimated TPM, the observed and expected cell prevalence of each state is shown in Figure 5.

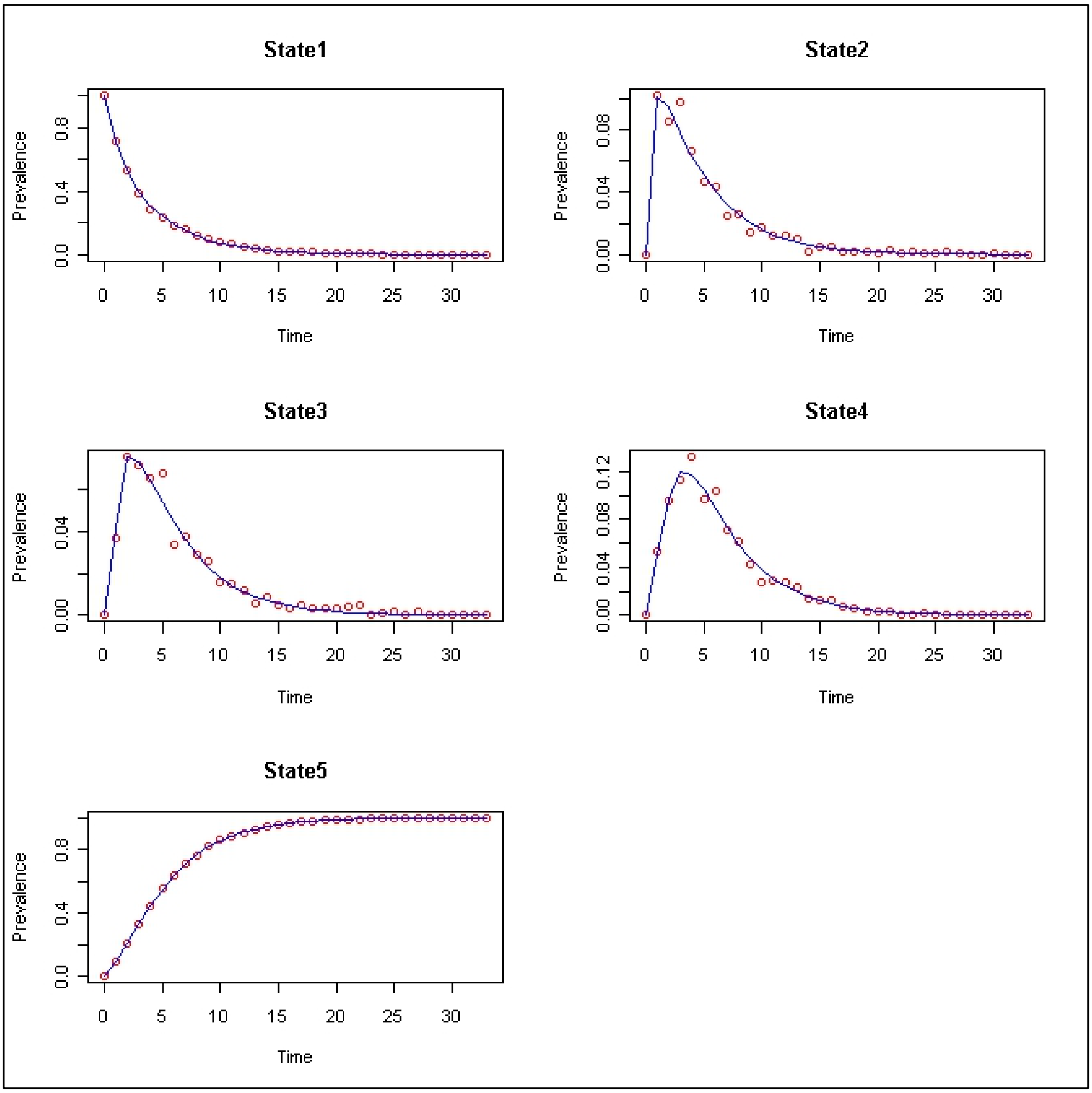

Figure 5: Observed and expected state prevalence for process with parameters in Equation (91) with $n = 1000, T = 50$. Red dots: Observed prevalence, blue lines: Expected prevalence

Figure 5 indicates that the expected state prevalence provides a good fit to the observed state prevalence. We also see that it was not necessary to observe the process for more than 30 time units, since most processes reached the absorbing state by then.

The average time for each simulation to move into the absorbing state since starting the process was 6.153 time units with an interquartile range of $8 - 3 = 5$. The sample mean time to enter the absorbing state is thus close to the 6.07 calculated for the true process parameters. Next, the process was simulated with $T = 3$ and $T = 8$ to assess the effect of observing a chain for time periods at which 25% and 75% of the chains are expected to have reached the absorbing state respectively. The number of chains simulated, n, was set to 1000, 500 and 100 for these simulations so that the effect of sample size on the estimates could also be determined. The estimated TPMs for these values of T are shown in Table 11. The SSD's for all estimated parameters are shown in Table 12, sorted by increasing SSD's.

n	$T = 3$	$T = 8$
1000	$\hat{P} = \begin{bmatrix} 0.705 & 0.099 & 0.051 & 0.045 & 0.099 \\ 0.107 & 0.198 & 0.385 & 0.166 & 0.144 \\ 0.053 & 0.084 & 0.084 & 0.481 & 0.298 \\ 0.021 & 0.071 & 0.114 & 0.357 & 0.436 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\hat{P} = \begin{bmatrix} 0.713 & 0.102 & 0.041 & 0.049 & 0.094 \\ 0.117 & 0.179 & 0.362 & 0.222 & 0.119 \\ 0.077 & 0.046 & 0.082 & 0.503 & 0.292 \\ 0.049 & 0.053 & 0.105 & 0.401 & 0.392 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
500	$\hat{P} = \begin{bmatrix} 0.699 & 0.108 & 0.052 & 0.040 & 0.1 \\ 0.125 & 0.205 & 0.375 & 0.116 & 0.179 \\ 0.063 & 0.127 & 0.079 & 0.460 & 0.270 \\ 0.031 & 0.092 & 0.092 & 0.308 & 0.477 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\hat{P} = \begin{bmatrix} 0.712 & 0.101 & 0.041 & 0.051 & 0.094 \\ 0.110 & 0.150 & 0.405 & 0.229 & 0.106 \\ 0.080 & 0.050 & 0.070 & 0.490 & 0.310 \\ 0.037 & 0.054 & 0.097 & 0.437 & 0.374 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
100	$\hat{P} = \begin{bmatrix} 0.703 & 0.110 & 0.073 & 0.009 & 0.105 \\ 0.182 & 0.273 & 0.273 & 0.136 & 0.136 \\ 0.071 & 0.286 & 0.071 & 0.500 & 0.071 \\ 0 & 0.3 & 0.1 & 0.4 & 0.2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\hat{P} = \begin{bmatrix} 0.738 & 0.099 & 0.047 & 0.039 & 0.078 \\ 0.137 & 0.137 & 0.431 & 0.196 & 0.098 \\ 0.059 & 0.137 & 0.118 & 0.373 & 0.314 \\ 0.034 & 0.034 & 0.103 & 0.448 & 0.379 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Table 11: Estimated TPM of process with parameters in Equation (91)

n	T	SSD
1000	50	0.00681108
1000	8	0.004270605
1000	3	0.01439214
500	8	0.01452945
500	3	0.03762921
100	8	0.04632581
100	3	0.2146737

Table 12: SSD for process with parameters shown in Equation (91)

Table 12 shows that for the values of T and n being compared, the parameters are estimated with more accuracy if the number of samples is increased, compared to increasing the time for which the process is monitored. Table 12 shows a better fit for $n = 1000$ when $T = 8$ compared to when $T = 50$ which is counterintuitive. Eventhough the estimates for $T = 8$ and $T = 50$ are relatively similar, the transition rate from state 3 to state 4 is estimated more accurately with $T = 8$ when compared to $T = 50$ and this difference in accruracy is big enough to make the SSD for $T = 8$ smaller than that of $T = 50$. It is expected that this pneomenon will not always happen in practise and it is therefore still desired to have T as large as practically possible, but we see that there may be a threshold value for T after which an increase of T beyond such a threshold will not lead to a great gain in accuracy. The observed and expected state probabilities for the process simulated with $n = 100, T = 3$ is shown in Figure 6 which indicates a good fit to prevalence in states 1,2,4 and 5 up to three time units. The plot is shown for $n = 100, T = 3$ to indicate the fit for the model with the greatest SSD.

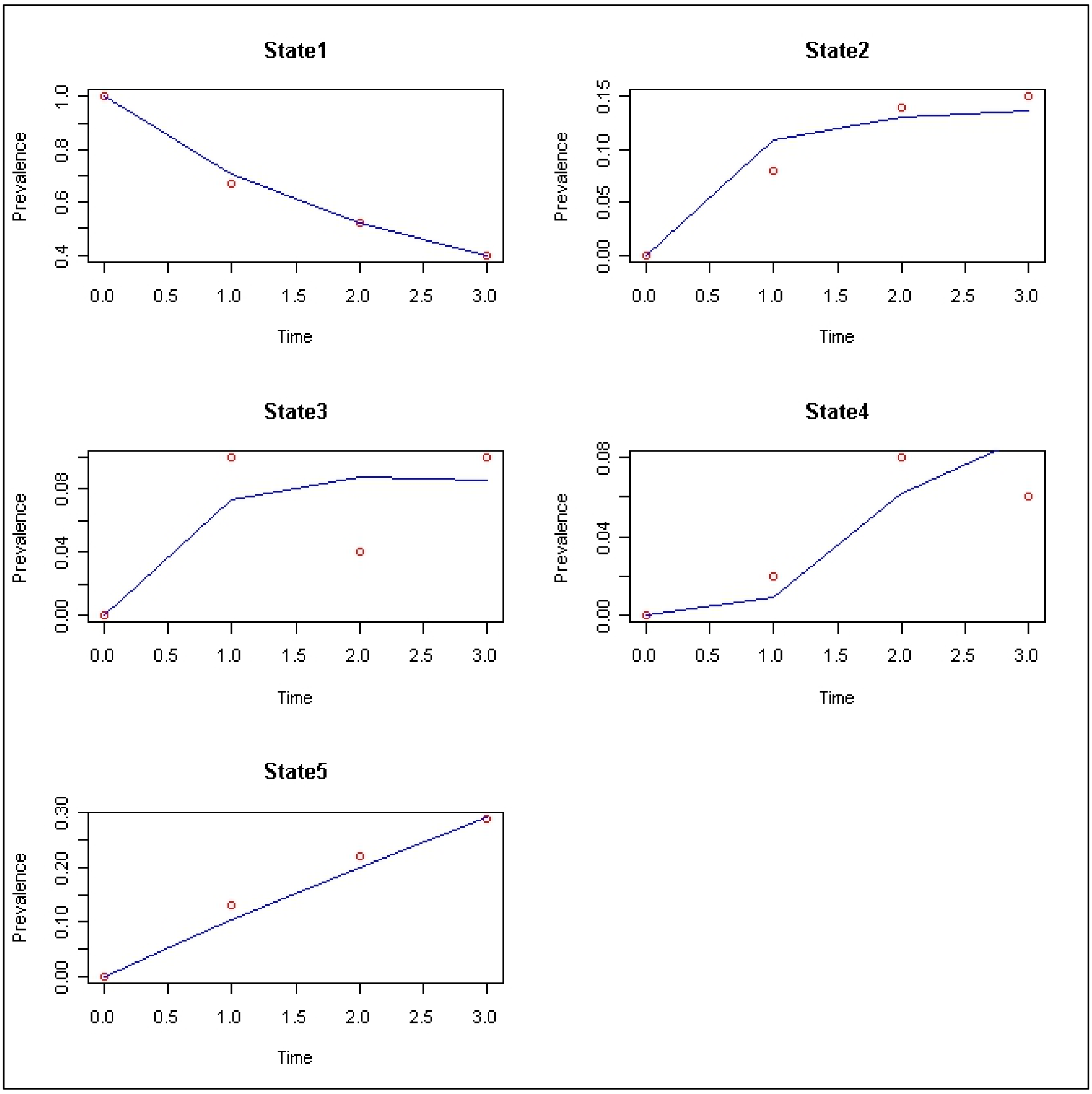

Figure 6: Observed and expected state prevalence for process with parameters in Equation (91) with $n = 100, T = 3$. Red dots: Observed prevalence, blue lines: Expected prevalence

A process with a low probability of entering one of the states was simulated to analyse the effect of T and n on the ability to provide robust estimates of the process parameters if one of the states has a low probability of being observed. The following process was simulated

$$P = \begin{bmatrix} 0.7 & 0.1 & 0.05 & 0.05 & 0.1 \\ 0.1 & 0.3 & 0.35 & 0.05 & 0.2 \\ 0.1 & 0.05 & 0.5 & 0.05 & 0.3 \\ 0.05 & 0.05 & 0.1 & 0.4 & 0.4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (93)$$

The probability to enter state 4 is low for the process described by Equation (93). The expected time from entering into state 1 until a move into the death state is 6.03 time units as estimated by Equation (9). The estimated TPM with $T = 50$ and $n = 1000$ is shown in Equation (94).

$$\hat{P} = \begin{bmatrix} 0.718 & 0.104 & 0.041 & 0.048 & 0.089 \\ 0.091 & 0.278 & 0.370 & 0.049 & 0.211 \\ 0.111 & 0.044 & 0.484 & 0.062 & 0.299 \\ 0.038 & 0.048 & 0.102 & 0.392 & 0.421 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (94)$$

The estimated TPM in Equation (94) provides a good fit to the TPM in Equation (93). The influence of less observations and a smaller observation time was investigated on the TPM in Equation (93). The estimated TPM for various values of T and n is shown in Table 13 and the SSDs are shown in Table 14.

n	$T = 3$	$T = 8$
1000	$\hat{P} = \begin{bmatrix} 0.711 & 0.095 & 0.053 & 0.046 & 0.095 \\ 0.114 & 0.275 & 0.358 & 0.041 & 0.212 \\ 0.105 & 0.056 & 0.457 & 0.043 & 0.340 \\ 0.020 & 0.040 & 0.140 & 0.320 & 0.48 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\hat{P} = \begin{bmatrix} 0.714 & 0.100 & 0.044 & 0.049 & 0.092 \\ 0.095 & 0.282 & 0.379 & 0.054 & 0.190 \\ 0.105 & 0.042 & 0.498 & 0.059 & 0.296 \\ 0.036 & 0.065 & 0.092 & 0.376 & 0.432 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
500	$\hat{P} = \begin{bmatrix} 0.711 & 0.107 & 0.053 & 0.039 & 0.089 \\ 0.123 & 0.298 & 0.316 & 0.053 & 0.211 \\ 0.107 & 0.080 & 0.387 & 0.040 & 0.387 \\ 0.045 & 0.023 & 0.091 & 0.364 & 0.477 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\hat{P} = \begin{bmatrix} 0.709 & 0.105 & 0.045 & 0.049 & 0.092 \\ 0.107 & 0.249 & 0.423 & 0.043 & 0.178 \\ 0.096 & 0.038 & 0.504 & 0.058 & 0.303 \\ 0.013 & 0.057 & 0.107 & 0.371 & 0.453 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
100	$\hat{P} = \begin{bmatrix} 0.698 & 0.108 & 0.086 & 0.014 & 0.095 \\ 0.190 & 0.238 & 0.286 & 0 & 0.286 \\ 0.10 & 0.15 & 0.45 & 0 & 0.30 \\ 0 & 0 & 0.25 & 0.75 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\hat{P} = \begin{bmatrix} 0.706 & 0.121 & 0.049 & 0.035 & 0.089 \\ 0.123 & 0.211 & 0.456 & 0.035 & 0.175 \\ 0.055 & 0.068 & 0.438 & 0.014 & 0.425 \\ 0.042 & 0 & 0.083 & 0.542 & 0.333 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Table 13: Estimated TPM of process with parameters in Equation (93)

Table 13 indicates that when 100 observations of the process with a small probability to enter state 4, is made, the estimated TPM suggests that it is not possible to enter state 4 from states 2 and 3 when $T = 3$. In this scenario it also seems like it is not possible to enter the absorbing state from state 4 since this was not observed. Table 14 indicates again that for the values of T and n considered, a smaller sample size has a greater influence on the accuracy of the estimators when compared to the observation time. The observed and expected state prevalence for the process is shown in Figure 7 which indicates that the estimated parameters provide a good approximation for the prevalence in states 1, 2, 3 and 5. Figure 7 fails to indicate the estimated parameters' inability to describe movements to and from state 4. This indicates that the observed and expected state prevalence plots as a measure of goodness-of-fit may not be lacking when data is scarce and the probability of entering some states are small.

n	T	SSD
1000	50	0.002887786
1000	8	0.003914031
1000	3	0.02025226
500	8	0.01406290
500	3	0.03176102
100	8	0.07185263
100	3	0.3536935

Table 14: SSD for process with parameters shown in Equation (93)

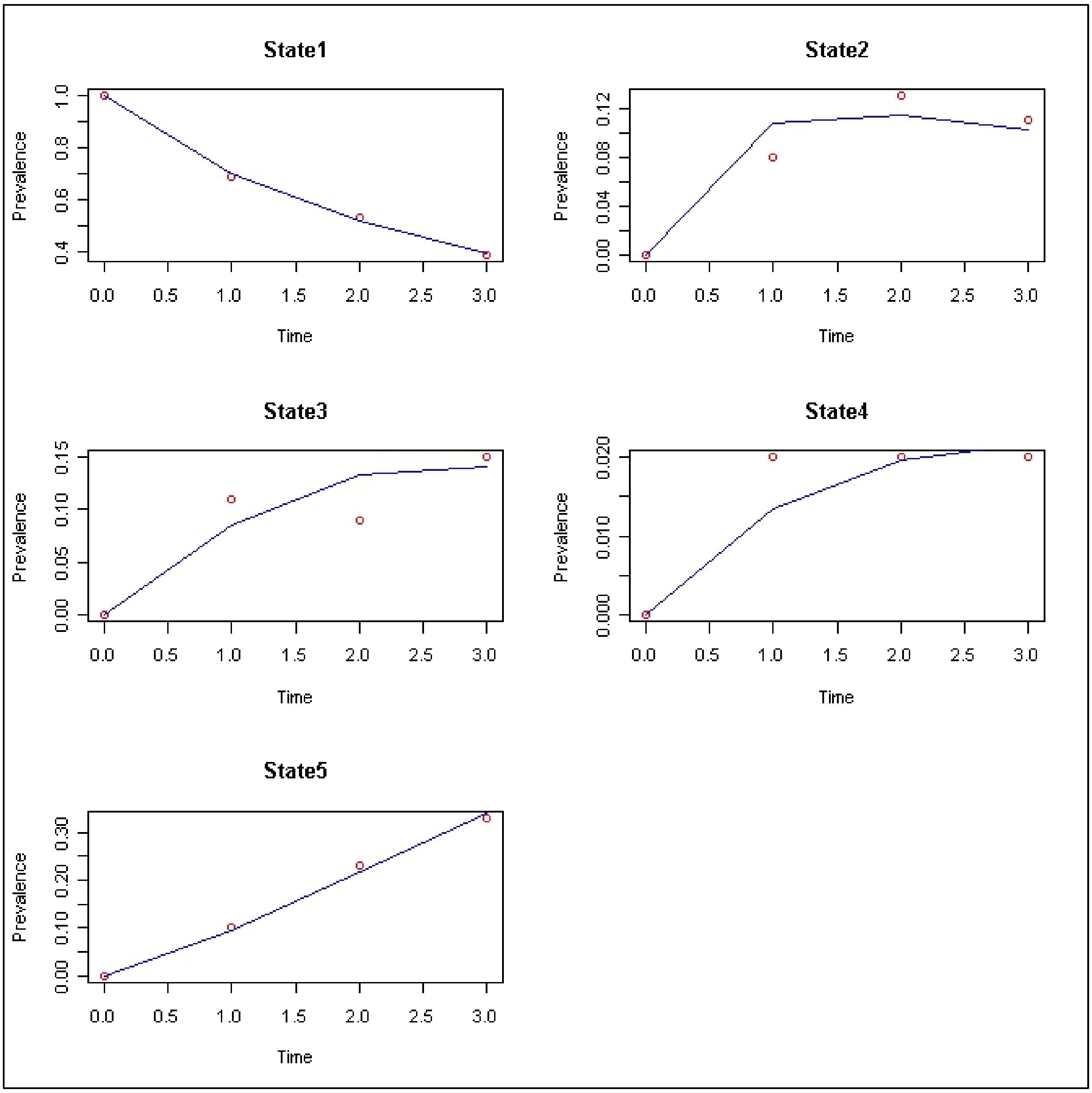

Figure 7: Observed and expected state prevalence for process with parameters in Equation (93) with $n = 100, T = 3$. Red dots: Observed prevalence, blue lines: Expected prevalence

4.3.2 Nonstationary Markov chain

Example VI

A nonstationary five state Markov process was implemented by assuming that the survival time in the first state follows a Weibull($\lambda = 0.2,\ \gamma = 2.5$) distribution. The Weibull distribution was used since it is often used in survival analysis and since it has a heavy tail. The parameters of the Weibull distribution were chosen so that the hazard function would increase exponentially over time to clearly show that treating this process as stationary would be incorrect. The hazard function for a Weibull distribution is given by

$$h(t) = \lambda \gamma t^{\gamma - 1}.$$

The Weibull hazard function for the parameters $\lambda = 0.2$, $\gamma = 2.5$ is plotted in Figure 8.

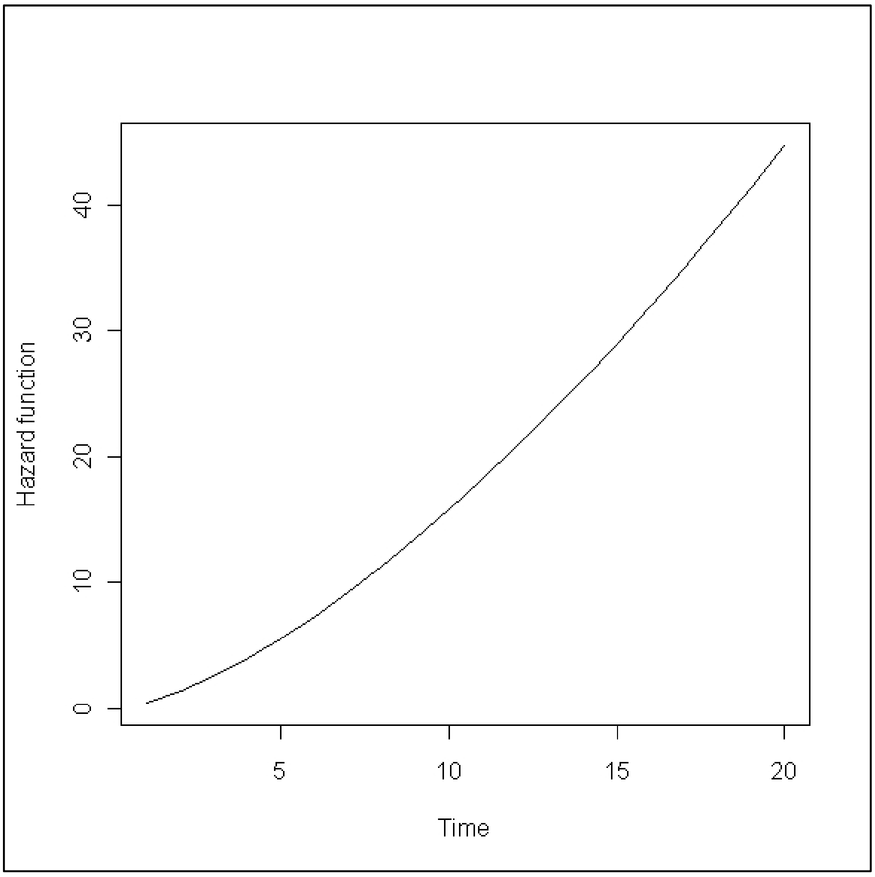

Figure 8: Hazard function for Weibull distribution with $\lambda = 0.2, \gamma = 2.5$

We calculate the probability to stay in state 1 at every time point as follows:

$$P_{1,1}^{(t,t+1)} = \frac{S(t+1)}{S(t)} = \frac{e^{-(\lambda(t+1))^{\gamma}}}{e^{-(\lambda t)^{\gamma}}} = \frac{e^{-(0.2(t+1))^{2.5}}}{e^{-(0.2t)^{2.5}}} = e^{-(0.2(t+1))^{2.5}+(0.2t)^{2.5}}. \tag{95}$$

The form of the hazard and survival function for the Weibull distribution was taken from Collet[18].

The transition probabilities from state 1 to states 2, 3 and 4 were all assumed to be a third of $0.8(1 - P_{1,1}^{(t,t+1)})$ so that $P_{1,5}^{(t,t+1)} = 0.2(1 - P_{1,1}^{(t,t+1)})$. All the transition probabilities from state 1 were therefore time dependent. The rest of the TPM was assumed to be the same as that of Equation (91). The system was simulated 1000 times with $T = 50$ and the nonstationary TPM was estimated using Equation (40). The nonstationary TPM estimator provides an estimated TPM for every time step. The "*list*" variable type in R was used to estimate the TPM at every time point. This variable type acts like a vector of which each element can be of a different type, including matrices.

The observed and expected state prevalence from the estimator is shown in Figure 9 which indicates that the estimated transition probabilities fit the process well.

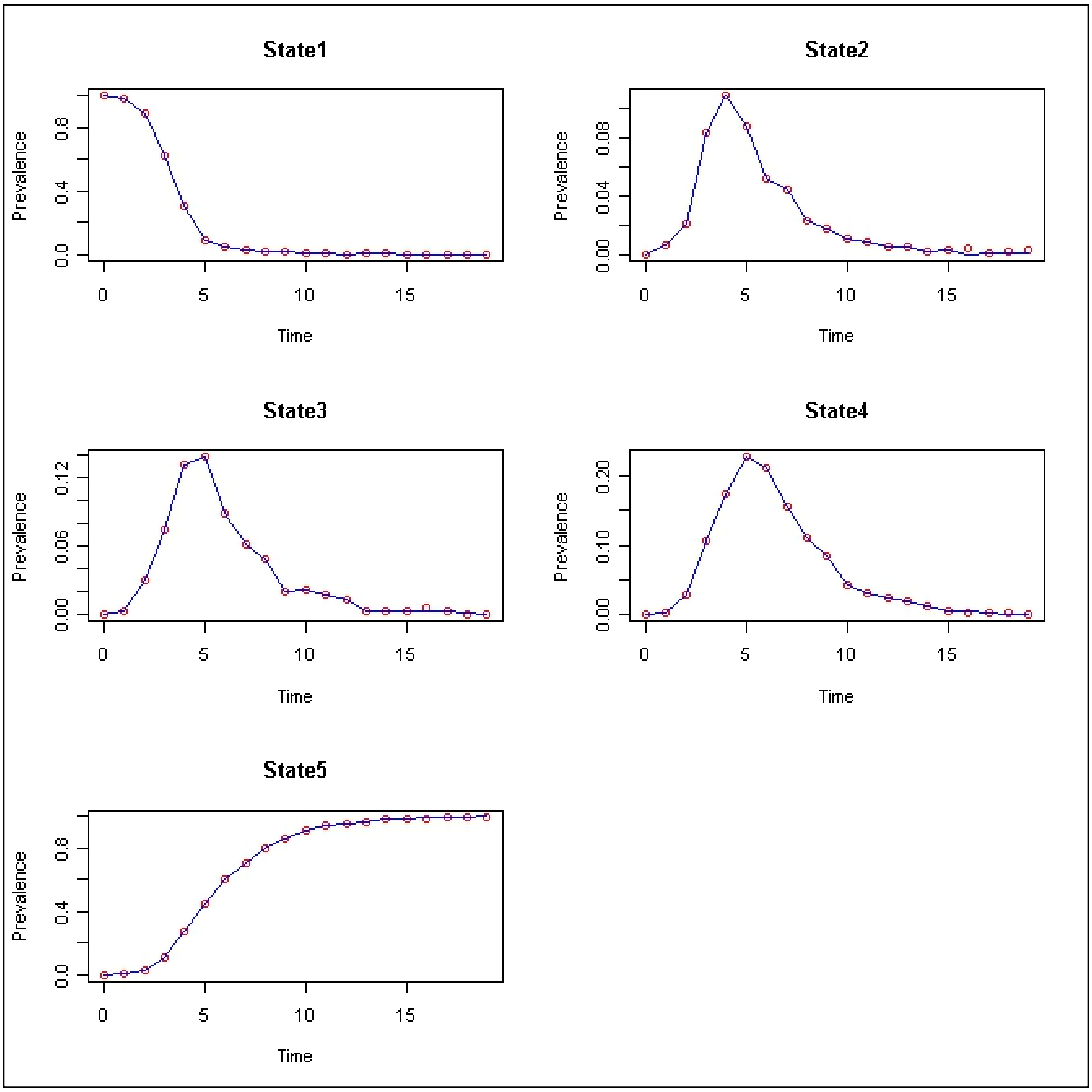

Figure 9: Observed and expected state prevalence of nonstationary Markov chain estimated by Equation (40). Red dots: Observed prevalence, blue lines: Expected prevalence

The nonstationary process was also estimated by ignoring the time dependence and using Equation (39) to estimate the process as if it is a stationary process. This was done to get an idea of the error one can make when assuming a process is stationary when the underlying process is nonstationary. The observed and expected state prevalence is shown in Figure 10 which indicates that treating the data as being from a stationary process fails to reproduce the observed state prevalence. We therefore see that the graphs of expected and observed state prevalence can be an indicator of whether a process should be treated as nonstationary or stationary.

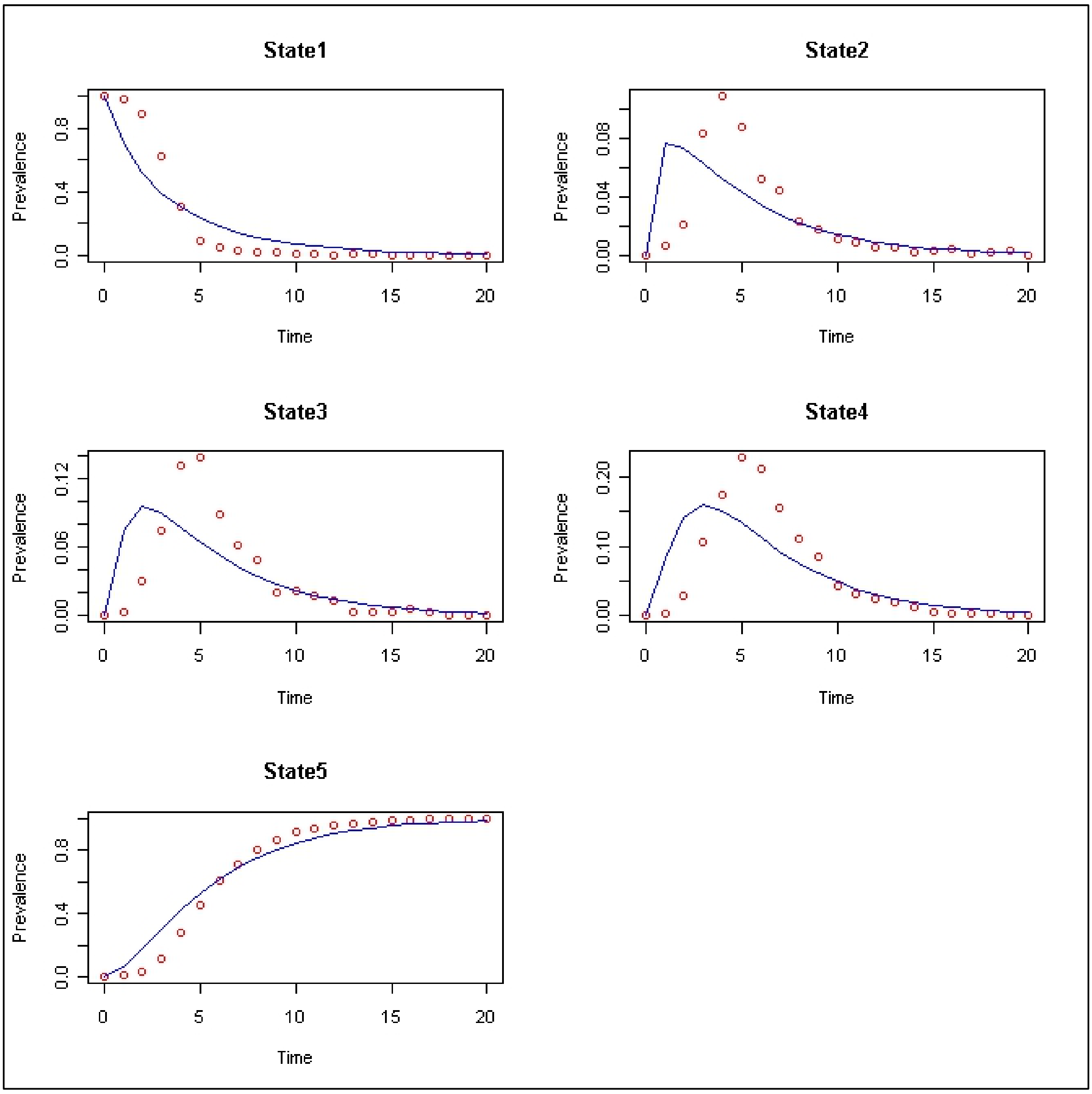

Figure 10: Observed and expected state prevalence of nonstationary Markov chain estimated by Equation (39). Red dots: Observed prevalence, blue lines: Expected prevalence

One would expect the ability of the Equation (39) to accurately estimate the TPM to decrease as the sample size decreases. This is because the TPM is calculated at every time point and a small sample size was already seen in Section 4.2 as a potential problem in estimating a TPM when it is estimated for all time periods in a sample.

The sample size was therefore reduced to 500 and 100 and the nonstationary TPMs were estimated. The observed and expected state prevalence for these two scenarios are shown in

Figure 11 and Figure 12 which indicates that the nonstationary estimator of the TPM produces a good fit to the observed data even when the sample size is reduced. It is still difficult to determine the fit of the estimates to the true data generation process and therefore the true state prevalence as calculated by the actual TPM was added to the plots as a green line. From this we see that the estimated process parameters provide a good fit to the data even with a sample size of 100, with the fit of the sample with 500 processes being better as one would expect. The number of time points was also reduced to assess the effect of this on the estimates. The observed and expected state prevalence plots are shown in Figure 13 and Figure 14 which indicates a good fit to the observed data. However, Figure 13 and Figure 14 also indicate the nonstationary estimator can over fit the data since the observed and expected state prevalence are the same, which may mislead one to think it is a perfect fit.

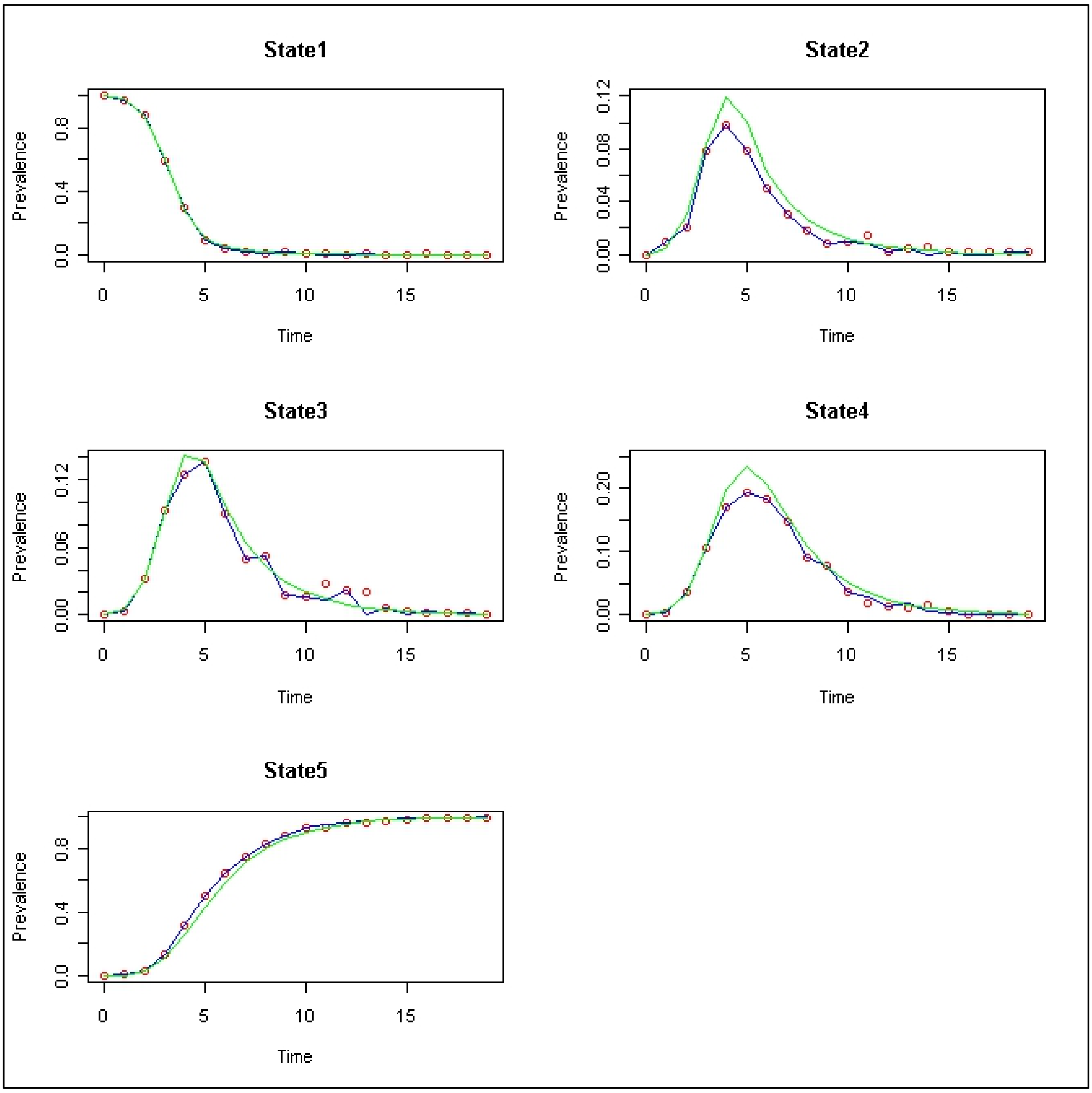

Figure 11: Observed and expected state prevalence of nonstationary Markov chain ($n = 500, T = 50$) estimated by Equation (40). Red dots: Observed prevalence, blue lines: Expected prevalence, green lines: true prevalence

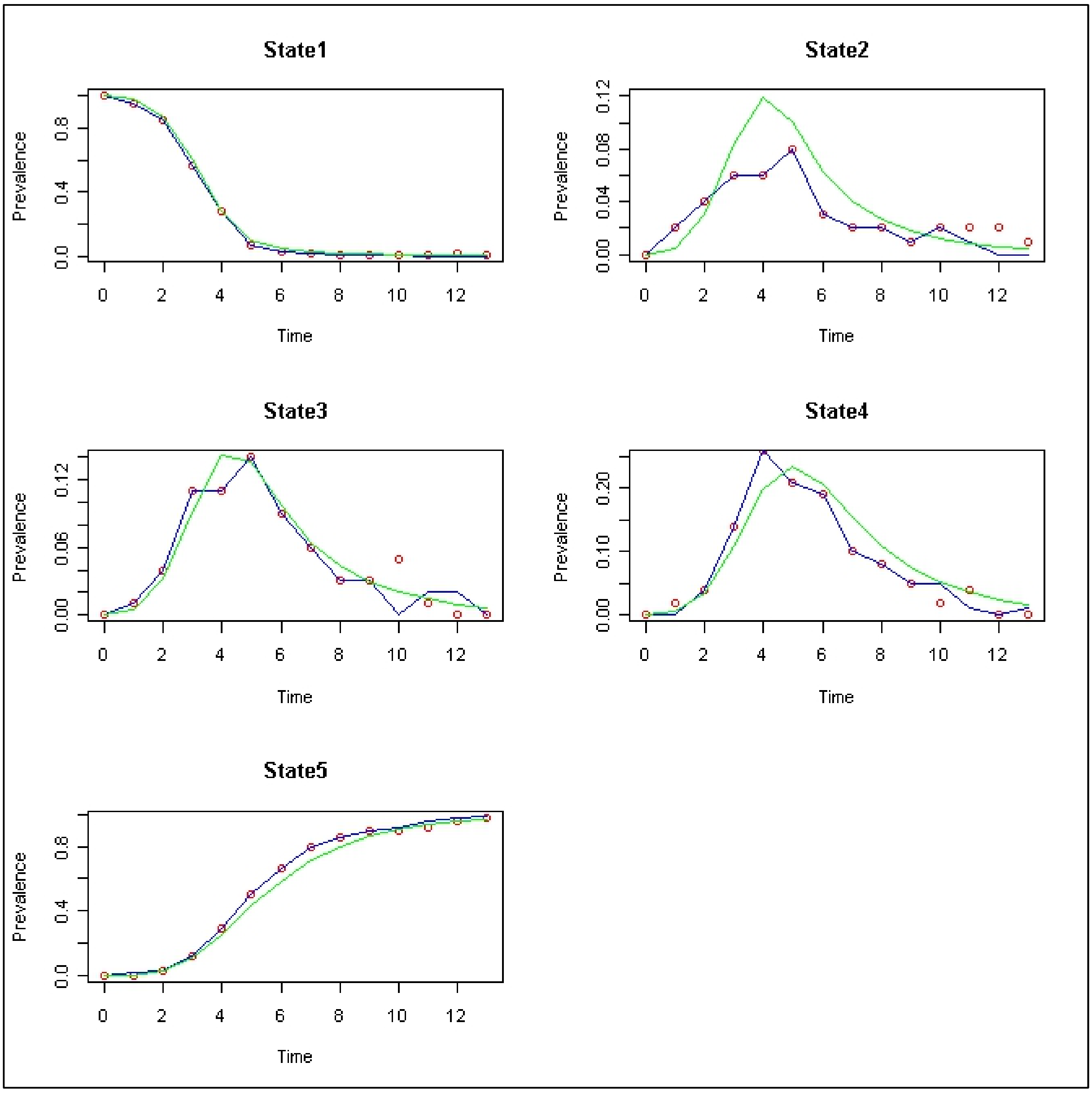

Figure 12: Observed and expected state prevalence of nonstationary Markov chain (n= 100, T= 50) estimated by Equation (40). Red dots: Observed prevalence, blue lines: Expected prevalence, green lines: true prevalence

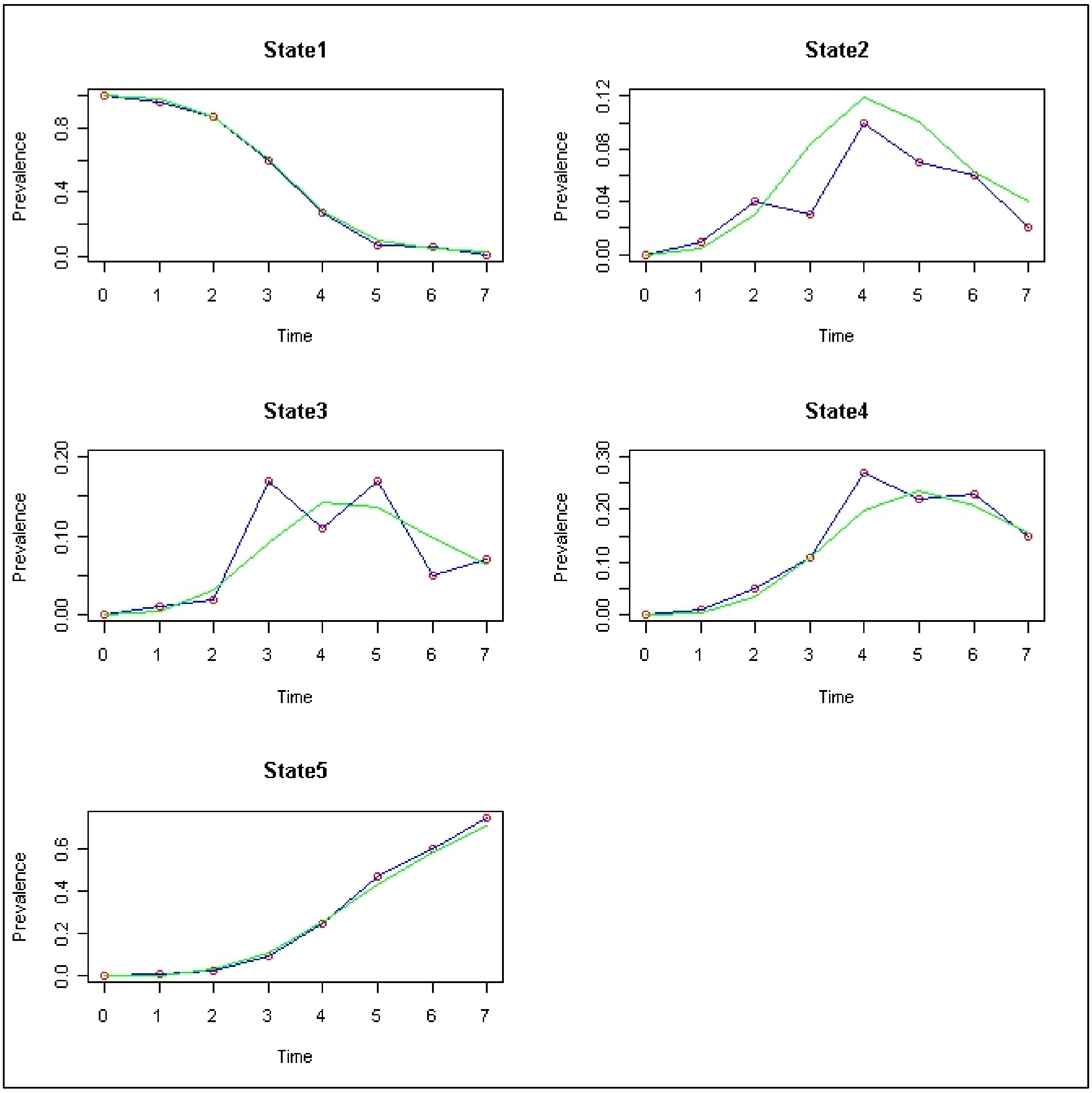

Figure 13: Observed and expected state prevalence of nonstationary Markov chain (n= 100, T= 8) estimated by Equation (40). Red dots: Observed prevalence, blue lines: Expected prevalence, green lines: true prevalence

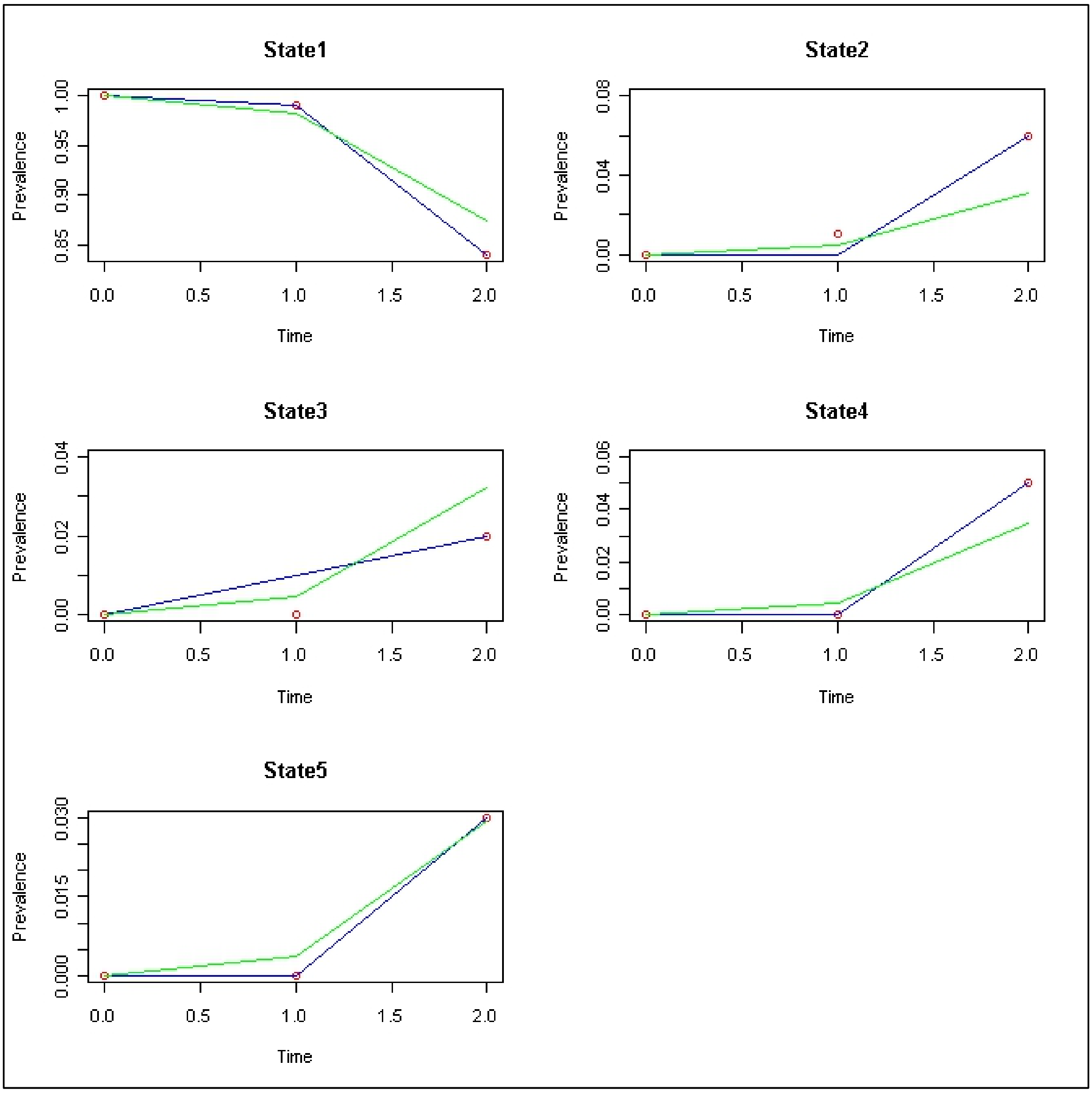

Figure 14: Observed and expected state prevalence of nonstationary Markov chain (n= 100, T= 3) estimated by Equation (40). Red dots: Observed prevalence, blue lines: Expected prevalence, green lines: true prevalence

4.4 Processes that are observed at the same irregular time points: The method of Kalbfleisch and Lawless

The ML estimator of Kalbfleisch and Lawless[53] was discussed in Section 3.5. Kalbfleisch and Lawless provided a quasi-Newton algorithm which maximises the log-likelihood function of a Markov jump process of which all observations were made at equal irregular time points. Kalbfleisch and Lawless provide an example in their article of a three state Markov jump process in which the smoking behaviour of children were observed at four time points. The states in the process were “child has never smoked”, “child is currently a smoker”, and “the child has smoked but has stopped now”. All the children in the sample started the process in state 1 and the process was observed times $t_1 = 0.15, t_2 = 0.75, t_3 = 1.1, t_4 = 1.9$. The TCM was provided at each of these observation times. The algorithm of Kalbfleisch was programmed in R and tested with the example provided by Kalbfleisch and identical estimated transition intensities were obtained.

Example VII

We extended Example I by simulating a Markov jump processes with transition intensity matrix in Equation (80) with $n = 1000$ and observation times at $t_1 = 0.5, t_2 = 1.2, t_3 = 2, t_4 = 2.5, t_5 = 3.1, t_6 = 4, t_7 = 5.1, t_8 = 5.7, t_9 = 6.9$ and all processes starting in state 1. These observation times were chosen such that that they are not equidistant and so that $T = 6.9$ which is beyond the average time in takes the process to reach the absorbing state (4.97 time units as shown in Example I).

The algorithm of Kalbfleisch considers all non-zero off-diagonal entries of the transition intensity matrix as entries into a vector $\boldsymbol{\theta}$ which has to be estimated. For this example we will let $\boldsymbol{\theta} = (\lambda_1, \mu_1, \lambda_2, \mu_2)$ so that the transition intensity matrix is identical to Equation (83).

The algorithm of Kalbfleisch requires the partial derivates of $\boldsymbol{A}$ with respect to the elements of $\boldsymbol{\theta}$ which are shown below.

$$\frac{\partial \boldsymbol{A}}{\partial \lambda_1} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{A}}{\partial \mu_1} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{A}}{\partial \lambda_2} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{A}}{\partial \mu_2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

The algorithm was started with initial vector $\boldsymbol{\theta_0} = (0.2,0.15,0.15,0.25)$ which was chosen to be different from but close to the true process parameters, $\boldsymbol{\theta_{True}} = (0.15,0.2,0.2,0.2)$, as a first attempt to assess the ability of the algorithm to converge to $\boldsymbol{\theta_{True}}$. The values of the next two steps in the procedure described by Equation (69) are as follows

$$\boldsymbol{\theta_1} = (0.1487,0.2018,0.2001,0.2043)$$

$$\boldsymbol{\theta_2} = (0.1486,0.2010,0.2005,0.2031).$$

We see that $\boldsymbol{\theta_1}$ is already close to $\boldsymbol{\theta_{True}}$ and that $\boldsymbol{\theta_2}$ does not differ much from $\boldsymbol{\theta_1}$ which shows that the algorithm is successful in estimating $\boldsymbol{\theta_{True}}$ and does so within one step**.**

The effect of initial values on the algorithm was analysed by choosing $\boldsymbol{\theta_0} = (1,1,1,1)$. The next five steps in the algorithm are shown below,

$$\begin{aligned}
\boldsymbol{\theta_1} &= (-1.5989, -0.0754, -1.3697, -0.1521)\\
\boldsymbol{\theta_2} &= (-1.7886, 0.0728, -1.4208, -0.0072)\\
\boldsymbol{\theta_3} &= (-1.986, 0.2455, -1.4801, 0.1615)\\
\boldsymbol{\theta_4} &= (-2.1923, 0.4395, -1.5460, 0.3533)\\
\boldsymbol{\theta_5} &= (-2.4073, 0.6530, -1.6158, 0.5631).
\end{aligned}$$

We see that the $\boldsymbol{\theta_i}$ vectors do not converge when $\boldsymbol{\theta_0} = (1,1,1,1)$**.** The tenth iteration produced the following parameter estimate

$$\boldsymbol{\theta_{10}} = (-3.9586,2.5165, -2.2218,2.4206).$$

So it seems that the process is not converging. The eighteenth iteration produced

$$\boldsymbol{\theta_{18}} = (-400.2001,594.4464, -194.8836,583.8143)$$

and the $\boldsymbol{M}(\boldsymbol{\theta_{18}})$ matrix was singular which meant that $\boldsymbol{\theta_{19}}$ could not be calculated since the inverse of $\boldsymbol{M}(\boldsymbol{\theta_{18}})$ needs to be calculated. We therefore see that the initial value can influence the estimated process parameters when the Kalbfleisch and Lawless estimator is used.

If we calculate the initial $(i,j)^{\text{th}}$ transition intensity by dividing the number of transitions from state i to j by the amount of time spent in state i (the same as the estimator by Albert[4] discussed in Section 3.3), we get $\boldsymbol{\theta_0} = (0.1125,0.1864,1452,0.1866)$ with the next three steps being

$$\boldsymbol{\theta}_1 = (0.1471, 0.2009, 0.1976, 0.2032)$$

$$\boldsymbol{\theta}_2 = (0.1486, 0.2010, 0.2005, 0.2031)$$

$$\boldsymbol{\theta}_3 = (0.1486, 0.2010, 0.2005, 0.2031).$$

We see that the third and second iterations produced the same estimates indicating that the parameters converged quickly. We also see that the estimated parameters are close to $\boldsymbol{\theta}_{True}$. Similarly to Example III and Example IV, we see that using the estimator of Albert as initial values for the Kalbfleisch estimator may help when estimating the transition intensities.

4.5 Estimator by Kay and Jackson

Kay[56] considered a ML estimator of a Markov jump process where all processes are observed at possibly different irregular time points. The methods by Kay was specifically focussed at dealing with censored transition times states. Censored states can occur if not all state transitions are observed. In the example of Klotz and Sharpless[57] we saw for example in Table 2, that the patient with PatID = 5 was in state 1 at time t_6 and in state 3 at time t_8 but we do not know if the patient had moved to state 2 in between times t_6 and t_8, nor do we know the exact time at which the patient moved to state 3. This is typical if a process is not observed exactly when state transitions occur. The method of Kalbfleisch and Lawless[53], discussed in Section 3.5 and implemented in Example VII, can be used if all processes are observed at the same time points, even if the time points were not equidistant. The method of Kay extends this by relaxing the assumption that all processes have to be observed at the same time point.

Equation (71) shows the likelihood function for every “alive” state observed. The estimator of Kay states that one expects the death of a patient to be recorded within a day and therefore uses Equation (72) as the likelihood function which incorporates all the possible states to which a patient could move from their last observed alive state and then dying within one day from that state.

A stationary three state Markov jump process will be investigated with the effect of using a TPM in the likelihood function being compared with that of using Equation (79) in the likelihood to calculate transition probabilities in the likelihood function.

Example VIII

A Markov jump process similar to Example I was simulated with the observation times being randomly chosen for every patient from an $Unif(0.5,1.5)$ distribution so that the expected value of the time between observations would be one time unit with a range of 0.5 to 1.5 time units. One time unit is thought to represent one year in this example and so the (1) time unit in $p_{X_i^{(h)},X_s}(1)$ in Equation (72) was specified to be $\frac{1}{365} = 0.00274$. Equation (80) was used as the true transition intensity matrix in the simulation with $n = 1000$ and $T = 50$. The *nlm* function was used to optimise the likelihood function and the initial values of the iterative process was set equal to the number of transitions from state i to j divided by the amount of time spent in state i (the same as the ML estimator based on exactly observed state transitions discussed in Section 3.3). When using the default values for *nlm* function we did not estimate transition rates which seemed reasonable. This problem was overcome by specifying that the difference in successive values of the parameter estimates do not exceed 0.1 units by specifying the option *stepmax=0.1* in the *nlm* function, so as to prevent big jumps in successive estimates.

The estimated parameters are provided in Table 15 for the scenario where a TPM is used in the likelihood function (Scenario 1) and where Equation (79) is used to calculate transition probabilities in the likelihood (Scenario 2). The computation time in seconds and SSD are also shown in Table 15.

Scenario	**Estimated transition intensity matrix**	**Computation time (seconds)**	**SSD**
1	$\hat{A} = \begin{bmatrix} -0.3062 & 0.1107 & 0.1955 \\ 0.1475 & -0.3671 & 0.2196 \\ 0 & 0 & 0 \end{bmatrix}$	36.41	0.004700923
2	$\hat{A} = \begin{bmatrix} -0.4216 & 0.2065 & 0.2151 \\ 0.1467 & -0.3454 & 0.1987 \\ 0 & 0 & 0 \end{bmatrix}$	12.74	0.0062628

Table 15: Estimated transition intensity matrices based on estimator by Kay with Equation (72) as likelihood function. Scenario 1 uses the TPM in the likelihood function and Scenario 2 uses Equation (79) in the likelihood function

Table 15 indicates that using the TPM in the likelihood function provides more accurate estimates of the transition intensity matrix compared to using Equation (79). Using the TPM in the likelihood function does however require more computation time (almost three times more). We see that transition rate from state 1 to state 2 was estimated with the least accuracy in both scenarios.

In the likelihood function of Jackson, the timing of patients that die are assumed to be recorded exactly and therefore proposes that one should use Equation (73) as likelihood function which differs from Equation (72) in that the transition intensity from the state before death is used instead of the transition probability in one day. The influence of this was investigated by using the same simulated data to generate Table 15, but with Equation (73) being used as likelihood function instead of Equation (72). We estimated the same transition intensity matrices as in Table 15 but with the computation time of optimising the likelihood function of Jackson being 10.71 and 12.54 seconds compared to 36.41 and 12.74 for the likelihood function of Kay for scenarios 1 and 2 respectively.

We therefore see in this example that there is not a difference in the accuracy of estimated transition intensities when assuming the transition probability from an alive state to dead in one day or using the transition intensity from an alive state to death, but that there is a gain in computation time.

Example IX

The estimated transition intensity matrices in Table 15 were based on the assumption that the time of death will be recorded within one day with one time unit in the simulation representing one year. The effect of observing transitions to the death states within one day was investigated by simulating the same process as in Example VIII but without transitions to the death state being observed within one day. The same methodology was used for implementing the *nlm* function in R as in Example VIII. Equation (71) was used as the likelihood function so that all observations were treated as the same and the difference in using the TPM in the likelihood compared to using Equation (79) was again assessed in terms of accuracy and computation time. The estimated transition intensity matrices of this approach is shown in Table 16 with scenario 1 representing the approach were the TPM is used in the likelihood function and scenario 2 representing the case where Equation (79) is used in the likelihood function.

Scenario	Estimated transition intensity matrix	Computation time (seconds)	SSD
1	$\hat{A} = \begin{bmatrix} -0.3975 & 0.1905 & 0.2071 \\ 0.1523 & -0.3568 & 0.2045 \\ 0 & 0 & 0 \end{bmatrix}$	58.36	0.003983
2	$\hat{A} = \begin{bmatrix} -0.2161 & 0.0635 & 0.1526 \\ 0.2054 & -0.4484 & 0.2431 \\ 0 & 0 & 0 \end{bmatrix}$	4.48	0.011615

Table 16: Estimated transition intensity matrices based on estimator by Kay with Equation (71) as likelihood function. Scenario 1 uses the TPM in the likelihood function and Scenario 2 uses Equation (79) in the likelihood function

We see again that using the TPM in the likelihood function provides more accurate estimates of the transition intensities compared to using Equation (79), with the former method taking more than 10 times the computation time than the latter method. When comparing Example VIII and Example IX we see that observing transitions to the death state within one day does not provide more accurate estimates than not observing such transitions exactly. This difference in accuracy is small when using the TPM in the likelihood function (Scenario 1 of Example VIII compared with that of Example IX) and much bigger when using Equation (79) in the likelihood function (Scenario 2 of Example VIII compared with that of Example IX).

Example X

The transition intensity matrix was estimated with the *msm* function in the *msm* package on the simulated data of Example VIII where transitions to the death state were observed exactly. When transitions to the death state are observed exactly, this is stated in the *msm* function as *death=3* (because the death state was state 3). The estimated transition intensity matrix is shown as Scenario A in Table 17.

The simulated data of Example IX where transitions to the death state were not necessarily observed exactly were also used in the *msm* function to estimate the transition intensity matrix. This is shown as Scenario B in Table 17. The computation time and SSD for scenarios A and B described above are also shown in Table 17.

Scenario	Estimated transition intensity matrix	Computation time (seconds)	SSD
A	$\hat{A} = \begin{bmatrix} -0.3465 & 0.1514 & 0.1951 \\ 0.1977 & -0.4151 & 0.2174 \\ 0 & 0 & 0 \end{bmatrix}$	6.28	0.000332
B	$\hat{A} = \begin{bmatrix} -0.3408 & 0.1474 & 0.1934 \\ 0.1900 & -0.4126 & 0.2226 \\ 0 & 0 & 0 \end{bmatrix}$	6.44	0.000661

Table 17: Estimated transition intensity matrices with the *msm* function based on observations for which the transition to the death state was observed exactly (Scenario A) and for data for which the transition to the death state was not necessarily observed exactly (Scenario B)

Table 17 indicates that the *msm* function provides more accurate estimates when transitions to the death state are observed exactly (Scenario A) compared to the case where such transitions are not observed exactly (Scenario B). The difference in computation time is small between Scenarios A and B with the transition intensities of Scenario A being estimated quicker. When comparing Table 17 with Table 15 and Table 16 we see that the *ms*m function is more accurate and time efficient in estimating the process parameters compared to the likelihood functions we programmed in R. Furthermore we see that the *msm* is more successful in using data in which the transitions to the death are observed exactly compared to the functions we programmed in R. We suspect that the author of the *msm* package has invested great effort in methods that more successfully optimise the likelihood functions than those used by us. Methods for more effectively optimising the likelihood function should be an interesting area of investigation for future research.

4.6 Discussion

We saw that the methods of Albert[4] and Tuma et al.[104] are accurate in estimating the transition intensities of a Markov jump process when state transitions are observed exactly even for a small sample. We saw that the accuracy of the estimators increases more by increasing the observation time compared to increasing the sample size. When comparing the methods wherein state transitions are observed exactly with that of equidistant time points we saw that there is not a big increase in accuracy for processes with three states where all states have relatively big chance of being visited. For processes with more states of which one of the states has a small probability of being occupied, much information is lost if the process transitions are not observed exactly.

When including covariate effects in a model, it is always important to use several starting values for all parameters being estimated to get an idea of the effect of starting values on the estimates. We described an approach for obtaining initial values in Example III and Example IV whereby several values for the initial values were used and then “reasonable” values were used as initial values for covariate effects. In both these examples we saw that using zero as an initial value in the *nlm* method would provide good estimates of the true covariate effects. The default starting value in the *msm* package for starting values of covariate effects is zero and this may be a good value to use when estimating covariate effects. It is still important to investigate the effects of other initial values on the estimates.

For a stationary Markov chain, we see that process parameters may not be estimated accurately if the sample size is small and/or the probability of the process to enter a specific state is small.

The nonstationary Markov chain estimator can be seen as a non-parametric type of distribution since no assumptions are made on the structure of the transition probabilities. This works well only if one is interested in interpolating state probabilities, but it is not possible to extrapolate state prevalence. Prior knowledge about the shape of nonstationary transition probabilities may be useful if one wants to extrapolate state prevalence.

The observed and expected state prevalence plots can be used to suggest if a nonstationary process should be used when a process was wrongfully treated as being stationary. These plots do however not always provide a robust view of how well the true process parameters were estimated and can be misleading when the nonstationary TPM estimator of Markov chain is implemented.

The method of Kalbfleisch and Lawless[53] provided accurate estimates of the process parameters and converged quickly in some cases but is sensitive to the starting values in the algorithm. We saw that using initial values that are similar to the method of Albert[4] produced accurate estimates of the process parameters. The ability of the method by Kalbfleisch and Lawless to estimate the process parameters depends on the possibility of calculating eigenvalues and eigenvectors of the transition intensity matrix for the estimated parameters at each step of the algorithm. This may produce problems, since the transition intensity matrix in Equation (82) for example has complex eigenvalues resulting in transition estimates that are complex numbers.

The methods of Kay[56] and Jackson[49] to estimate transition intensities worked well and we saw that it was more effective to use the TPM in the likelihood function compared to Equation (79), but that it is computationally more intensive. We indicated that the methods of Kay and Jackson to include the information of transitions to the death state being recorded with one day or exactly produced the same estimates for the transition intensities. We obtained similar estimates for the transition intensities when compared to the *msm* package, but the *msm* package produced more accurate estimates, is less computationally intensive and uses the information of deaths being recorded exactly more effectively than our attempts. We therefore encourage the use of the *msm* package.

Nonstationary Markov jump processes were not estimated as this falls outside the scope of this dissertation and may be considered for future research. As suggested by Kalbfleisch and Lawless[53] and Ocana-Riola[81], several transition intensity matrices can be considered over time periods for which a process is stationary. One could also include the effect of time on transition rates as a covariate effect as described in Section 2.4. In progressive models, one could fit a parametric, or Cox-type survival model to the event times which can be chosen to be nonstationary. Fitting nonstationary models to processes that are not progressive and for which the transition intensity matrices cannot be split into stationary intensity matrices may be troublesome. The users of such models may want to start by trying to model the diagonal entries of the transition intensities as parametric hazard functions and considering the off-diagonal entries as constant with one of the entries estimated as the difference between one and the sum of the other entries in a row. This will most likely be a non-trivial process involving a trial and error based approach to find the best hazard function for each state.

We describe one final example in which we simulated a five state process with covariate effects and where each process is observed at potentially different irregular time points. This is done to summarise all the observation and process possibilities described. The example will be conducted with the sole use of the *msm* package since we indicated that it is a very efficient package and we would like to assess how well it can estimate the parameters of a more complex situation than described in any of the examples above.

Example XI

A Markov jump process with five states was simulated with observations times being different for all processes and randomly chosen from a $Unif(0.5,1.5)$ distribution. An age and gender effect similar to that of Example IV was assumed for the transition intensity from state 1 to state 2. The remaining entries in the transition intensity matrix are shown in Equation (96). The transition from state 1 to state 1 is calculated as one minus the sum of the other transition intensities in row 1 of the matrix as is therefore indicated as # in Equation (96). The transition from state 1 to state 2 was simulated with $\beta_{12,0} = 0.2, \beta_{12,1} = 0.4, \beta_{12,2} = 0.05$ and $n = 1000$ process were simulated with $T = 50$. The age of patients were simulated from a $Unif(0,100)$ distribution and gender was simulated from a $Bernoulli(0.7)$ distribution with 1 representing females and 0 representing males.

$$A = \begin{bmatrix} \# & e^{\beta_{12,0}+\beta_{12,1}Sex+\beta_{12,2}Age} & 0.8 & 0.3 & 0.1 \\ 0.8 & -1.2 & 0.2 & 0.1 & 0.1 \\ 0.4 & 0.9 & -1.6 & 0.2 & 0.1 \\ 0.2 & 0.4 & 0.3 & -1.7 & 0.8 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{96}$$

The process parameters were estimated with the use of the *msm* function and by specifying the covariates effects and that transitions to the death state were observed exactly. The estimator of Albert[4] was used for initial values of the transition intensities with the initial values of the covariate effects set to zero. It took 18.5 minutes to for the *msm* procedure to estimate the parameters. The estimated transition intensity matrix is shown in Equation (97).

$$\hat{A} = \begin{bmatrix} -0.6714 & e^{0.1564+0.1707Sex+0.02388Age} & 0.2207 & 0.1787 & 0.1157 \\ 0.8746 & -1.244 & 0.168 & 0.0665 & 0.1349 \\ 0.1838 & 0.4616 & -0.9113 & 0.0813 & 0.1846 \\ 0.1848 & 0.2471 & 0.0807 & -1.027 & 0.5145 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{97}$$

When comparing Equations (96) and (97) we see that the transition intensities from state 1 to 3, from state 3 to 1, from state 3 to 2, from state 4 to 2 and from state 4 to 5 were not estimated accurately, but the other transition intensities were estimated accurately. We also see that the covariate effect of gender was underestimated and similarly for age. We therefore see that all transition intensities and the covariate effects are not always estimated accurately.

The standard error of estimates generated by the *msm* package is derived from the hessian matrix of the optimisation method and this matrix needs to be positive definitive to provide standard errors. For simulated data, the *msm* function could not provide standard errors for the covariate effects since the hessian matrix was not positive definite and we can therefore not comment on the significance of the covariate effects.

We used the *plot.prevalence.msm* function in R to plot the observed and expected state prevalence and this is shown in Figure 15. The length of stay in each state based on the estimated parameters in R can be calculated with the *totlos* function. From the sample we calculated that the expected number of time units that the process is in transient states is 6.36 time units and the estimated time in the transient states is 6.67 time units.

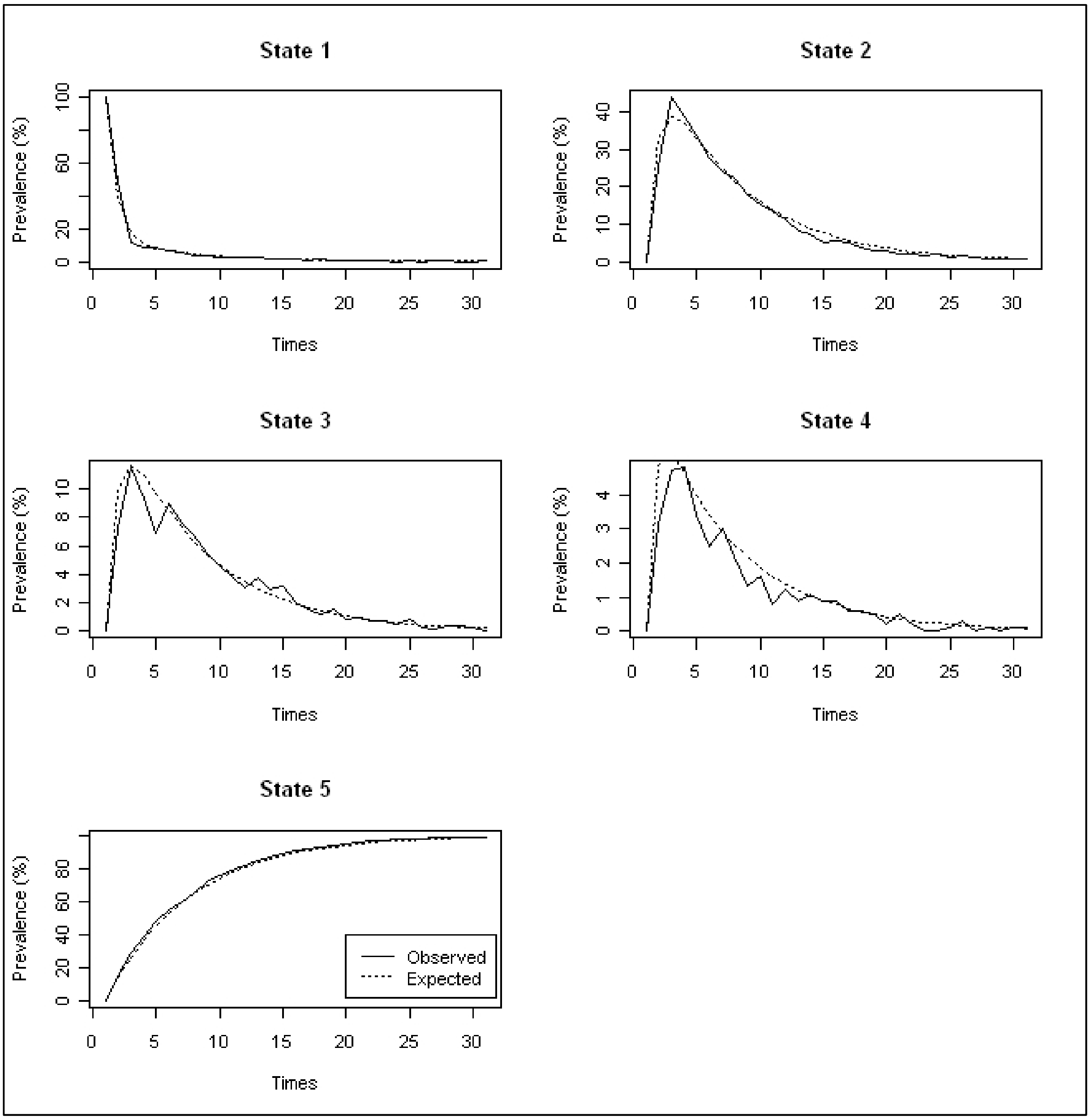

Figure 15: Observed and expected state prevalence for Example XI

Figure 15 indicates that the process estimates state prevalence well and we also saw that the number of time units in the transient states were estimated accurately. Figure 15 fails to indicate that the transition intensities from state 1 to 3, from state 3 to 1, from state 3 to 2, from state 4 to 2 and from state 4 to 5 were not estimated accurately.

We used the *pearson.msm* function with the *boot=TRUE* option to determine if the process provides a good fit to the data but the procedure could not calculate p-values. We therefore see that it is not always possible to use the Pearson-like test discussed in Section 3.7 to assess goodness-of-fit.

4.7 Summary

We compared various methods for estimating the parameters of Markov processes and indicated that the *msm* package of R is a useful tool in estimating process parameters. When simulating a Markov jump process with five states and covariate effects and where all processes are observed at possibly different time points we saw that methods discussed may not always be able to make inference on covariate effects, but are useful in estimating state prevalence and the number of time units the process is in transient states. This means that Markov processes should be able to provide good estimates of life expectancy and the number of people in each stage of a disease over time so that the morbidity and potential financial effects can be estimated. The methods described also enable the use of censored data to estimate the parameters of Markov processes.

Most of the methods discussed involve the estimation of the parameters of Markov jump processes, and Equation (24) can be used to calculate the appropriate TPM should one prefer to work with a Markov chain process.

Chapter 5: Conclusion

5.1 Introduction

Markov models are being used in health economics as a tool to predict the financial and societal impact of diseases and the treatment thereof. We have shown an example of how Markov models can be used to calculate the life expectancy and financial impact of a patient with a disease if such a disease is modelled as a Markov process. The primary focus of this dissertation was to investigate methods to estimate the parameters of such Markov models. The methods were discussed with special focus on disease models, but can be generalised to other stochastic processes modelled as Markov processes.

Maximum likelihood methods were discussed and implemented based on various types of observation schemes where the state of patients is recorded at various time points as longitudinal data. Tuma et al.[104] suggests that ML estimators are favourable to work with due to their good large sample properties, quantities derived from ML estimates are also ML estimates of the quantities and ML methods make it possible to include censored information.

We indicated that the likelihood function depends on the timing of observations and provided and implemented various likelihood functions based on different observations schemes. An in-depth discussion was provided for the theory behind estimating the parameters of processes for which the transitions between states are observed exactly (Section 3.3) and when the state of all processes in a sample are observed at fixed equidistant time points (Section 3.2). These methods have closed form expressions for the ML estimators and are easily calculated. The inclusion of covariate effects on transition intensities of processes where state transitions are observed exactly were discussed in Section 3.4. We also provided estimators for Markov processes where the state of all processes are observed at the same equidistant time points (Section 3.5) and when the state of each process is observed at possibly different and not equally spaced time periods. The methods of Sections 3.4, 3.5 and 3.6 do not provide closed form expressions for the process parameters and must be estimated using iterative optimisation functions such as the *nlm* function in R.

The methods were discussed under the assumption of parameters that are constant over time, i.e. stationary processes, with some suggestions being made on handling nonstationary processes in Section 3.8. Methods for handling censored transition times and states were also discussed.

The primary objective of this dissertation was not to give an in-depth discussion of goodness-of-fit methods and so two goodness-of-fit methods used in the *msm* package were discussed in Section 3.7 and implemented in Chapter 4. Tuma et al.[104] discusses an additional goodness-of-fit method that was not implemented in this dissertation and could be considered for future research. They also suggest that in addition to compared and expected transition prevalence one compare observed and expected number of movements between states and also the number processes that are in a specific state for different time periods simultaneously. We indicated that the Pearson-like goodness-of-fit method does not always provide indicative results and that the observed and expected state prevalence plots can be used to indicate if the assumption of a stationary process is viable. The observed and expected state prevalence plots do not however necessarily indicate if all the transition intensities are estimated accurately.

We indicated that the *msm* library of R created by Jackson[49] provides a powerful tool that practitioners can use when working using Markov processes in disease modelling.

There is a vast amount of literature available in the field of biostatistics on methods used in disease modelling and it was beyond the scope of this dissertation to provide a comprehensive discussion of all such methods. Section 5.2 provides a discussion of some methods used in estimating the parameters of Markov parameters which were not discussed in the dissertation. In Section 5.3 we provide an short discussion on alternative methods that can be used in disease modelling when the Markov processes described in the first four chapters of this dissertation are not valid. A thorough discussion of these methods are not provided, but the methods are merely mentioned to give the reader an idea of other methods that are available in the field of disease modelling.

The dissertation concludes with a summary in Section 5.4.

5.2 Literature not discussed

The methods and examples discussed in this dissertation were chosen such that it is not restricted to progressive models. This is because standard survival methods may be used for such diseases. Consider for example a model with $\{Well, Ill, Dead\}$ as state space such that a patient can move from the $Well$ state to the Ill state or the $Dead$ state and from the Ill state to the $Dead$ state but not back to the $Well$ state. As discussed by Meira-Machado et al.[73], Kaplan Meier and Cox type methods can be used to estimate the transition rate from the $Well$ state to the $Dead$ state by considering transitions to the Ill state as right censored observations.

A method was described for handling nonstationary processes when the processes are observed at equidistant time points and treated as a Markov chain process in Section 3.2. Other methods were mentioned for dealing with nonstationary data when the processes are treated as Markov jump processes. Hubbart et al.[47] mentions a method by which nonstationary processes are written as stationary processes following a time transformation. This method is not discussed in detail in the dissertation and the interested reader is refered to the article for further details. The incorporation of time dependent transition rates is a possible area for future research and has not received much attention as mentioned by Hubbart et al.

Alternative statistical software for the use of multi-state processes in disease modelling has been developed by Meira-Machado et al.[73] in the form of a package for R called *tdc.msm*[66]. This package can handle semi-Markov processes, time independent and dependent covariates in Cox type models and nonstationary Markov processes with piecewise constant parameters. Other software available includes a SAS algorithm written by Hui-Min et al.[48] and a SAS module created by Chang et al.[14] which can also be used for multi-state modelling of Markov processes. The algorithm of Hui-Min et al can handle stationary and nonstationary processes, and provide a 95% confidence interval for estimates but is not discussed in detail in this dissertation since it is only valid for progressive processes. The module of Chang et al.[14] can estimate covariate effects and state prevalence but cannot handle nonstationary data. The module uses the Pearson-type test for goodness-of-fit and also a cross-validation procedure where $\frac{2}{3}$ of the data is used to fit the parameters and the remaining $\frac{1}{3}$ is used to test the predictions. It is not clear from the description of Chang et al. if their module can handle processes that are not progressive.

A thorough investigation of the methods used in these software packages is not provided and is a possible topic for future research.

It was shown in Section 4.3 that when data is scarce or some states have a relatively small probability of being entered ML estimators can lead to inaccurate estimates. Furthermore, all the methods discussed in the first four chapters were based on longitudinal data, but this may not always be available. In some cases only the number of subjects in each state is known at each time point, but not the specific transition path of each subject. This type of data is called aggregate data, or macro data. In another article by Kalbfleisch and Lawless[53] which is not discussed in the dissertation a least squares estimator of transition intensities for aggregate data is provided.

Bayesian estimation methods also provide a possible solutions to handling aggregate data when prior information is available on the process parameters and this is discussed in Section 5.2.1.

5.2.1 Bayesian estimation methods

Billard[10] and Meshkani[75] are some of the most referenced contributors for Bayesian methods. Their work is however not specifically aimed at disease modelling and they assumed a matrix beta prior distribution which requires the assumption that none of the transition probabilities are zero. This implies that the models cannot have an absorbing state and therefore makes it irrelevant to disease modelling since disease models mostly have a death state.

Bayesian methods for aggregate data is discussed in a textbook entitled *Estimating the Parameters of the Markov Probability Model from Aggregate Time Series Data*[60] by Lee et al. In this textbook methods are discussed for aggregate data, but similar to the work by Billard et al.[10] and Meshkani et al.[75] a matrix beta prior distribution is assumed which is not applicable to disease models.

An overview of some of the Bayesian methods is provided below in the case where disease models are estimated without an absorbing death state. This could be the case for example where a disease for which a few possible events are possible but where the mortality rate following any of these events are negligible.

Since a matrix of parameters will be estimated in the Markov chain a prior distribution of a matrix will be used to estimate the posterior distribution of the matrix of parameters. One such prior distribution that can be used is the matrix beta distribution. For clarity the matrix beta distribution will be discussed below before commencing with the discussion of Bayesian estimation methods.

5.2.1.1 Matrix Beta distribution

The random row vector, $\boldsymbol{p}_i$ of dimension r, is said to have a multivariate beta distribution with parameters $(a_{i1}, a_{i2}, \dots, a_{ir})$, if the probability distribution function of $\boldsymbol{p}_i$ is given by

$$f(p_{i1}, p_{i2}, \dots, p_{ir}) = \frac{\Gamma(\sum_{k=1}^{r} a_{ik})}{\prod_{k=1}^{r} \Gamma(a_{ik})} \prod_{k=1}^{r} p_{ik}{}^{a_{ik}-1} \text{ where } \Gamma(r) = (r-1).$$

Now, if l independent row vectors, each of dimension r and with a multivariate beta distributions, are stacked underneath each other in a $l \times r$ matrix $\boldsymbol{P}$ we say the matrix $\boldsymbol{P}$ has a matrix beta distribution with parameters $A = [a_{ij}]$. The pdf of $\boldsymbol{P}$ is found by multiplying the pdf's of the l independent row vectors to obtain

$$f_{\boldsymbol{P}}^{(l,r)} = \prod_{i=1}^{l} \prod_{k=1}^{r} \frac{\Gamma(\sum_{k=1}^{r} a_{ik})}{\prod_{k=1}^{r} \Gamma(a_{ik})} p_{ik}{}^{a_{ik}-1}. \qquad (98)$$

The expected value of a matrix is equal to the expected value of its elements. Therefore the expected value of $\boldsymbol{P}$ is given in terms of its elements. The variance of the elements is also provided.

$$E(p_{ij}) = \frac{a_{ij}}{\sum_{j=1}^{r} a_{ij}} \qquad (99)$$

$$Var(p_{ij}) = \frac{E(p_{ij})[1 - E(p_{ij})]}{\sum_{j=1}^{r} a_{ij} + 1} \qquad (100)$$

5.2.1.2

5.2.1.3 TPM with matrix beta prior

Let's assume that the $\mathscr{s} \times \mathscr{s}$ TPM of a Markov chain, $\boldsymbol{P}$, has a prior matrix beta distribution with parameters $\boldsymbol{A} = [a_{ij}]$. We know that the likelihood function of n observations (in terms of the transition count matrix $\boldsymbol{F}$) is given by

$$L = \prod_{i=1}^{\mathscr{s}} \prod_{j=1}^{\mathscr{s}} p_{ij}^{f_{ij}}. \tag{101}$$

Now taking the product of Equations (98) and (101) and using only the terms involving p_{ij} the posterior distribution of $\boldsymbol{P}$ is proportional to

$$d(\boldsymbol{P}|\boldsymbol{A},\boldsymbol{F}) \propto \prod_{i=1}^{\mathscr{s}} \prod_{j=1}^{\mathscr{s}} p_{ij}^{f_{ij}+a_{ij}-1}.$$

Therefore we have that the posterior distribution of $\boldsymbol{P}$ is matrix beta with parameter

$$\boldsymbol{A}' = (\boldsymbol{F} + \boldsymbol{A}).$$

Under squared error loss the estimate of $\boldsymbol{P}$ is given by the expected value of the posterior distribution. The expected value of a matrix beta distribution is given in Equation (99).

Therefore the Bayesian estimator of the TPM is given by

$$\hat{p}_{ij} = \frac{f_{ij} + a_{ij}}{\sum_{j=1}^{r}(f_{ij} + a_{ij})}. \tag{102}$$

5.2.1.4 Implementing the Bayesian estimator

Equation (102) provides the Bayesian estimator which can be implemented if the FCM, $\boldsymbol{F}$, and the matrix of posterior parameters $\boldsymbol{A}$ are known. The FCM can be calculated from the observed data, but the value of $\boldsymbol{A}$ may be unknown. Billard et al.[10] provided a ML estimator for $\boldsymbol{A}$ when n FCM's have been observed in the past and a new FCM becomes available. The information from the old data (expressed as $(F_1, F_2, \dots, F_n)$ is then used to estimate $\boldsymbol{A}$ and the new FCM matrix is used as $\boldsymbol{F} \equiv \boldsymbol{F_{n+1}}$ in Equation (102).

Applying this to disease modelling could however be troublesome. There are two possibilities for accessing past data. The most obvious is that the FCM of n previously Markov chains for patients with similar characteristics is available. This will typically not be the case in clinical data due to the high costs of performing clinical trails, which results in predominantly new clinical trials being done everytime. Therefore if a new group of patients are being observed in a clinical trial, the TPM estimated for them will be unique and most often previous data will not be available.

The other possibility for past data is to consider the FCM for each patient (or randomly chosen groups of patients) separately and apply the techniques of Billard et al. This will be discussed next since the possibility follows directly from application of the methods in Billard.

Assume that n independent Markov chains with the same TPM are observed which corresponds to the health progression of n comparable patients up to time T. Now randomly divide the data into $k+1 \leq n$ subsets of patients. The $k+1$ subsets are used to be consistent with the notation provided by Billard et al. For each subset, calculate the FCM so that the following matrices are available: $(F_1, F_2, \ldots, F_k, F_{k+1})$ and then calculate the TPM matrix based on the methods of Billard et al with the $(n+1)^{\text{th}}$ sample corresponding to the $(k+1)^{\text{th}}$ subset of patients from the clinical trial. Repeat this procedure multiple times so that a bootstrap sample of TPM's is created. The average of bootstrap sample can then be used as an estimate of the TPM.

5.3 Alternative methods used in disease modelling

Markov processes are not always appropriate to model the disease progression of patients nor to answer all the questions a researcher may want to investigate in terms of disease modelling. It is therefore important to be aware of situations where Markov processes, as discussed in this dissertation, should be avoided.

The Markov property may not always be valid when transition probabilities depend on the history of patients. Meira-Machado et al.[73] mentions that there is not much literature available on the estimation of transition rates of non-Markov models but provides an overview of the literature that is available. They also mention that the Markov assumption can be validated by including a covariate representing the history of a patient and testing if this variable is significant. This can be done by using a binary variable to indicate if a patient was in a certain state previously. Processes for which the Markov assumption does not hold can be modelled by introducing more states which represent the history of patients. Consider for example a process with three states $Well, Ill,$ and $Dead$ where patients can move back to the $Well$ state after being in the Ill state. We expect the probability of moving to the $Dead$ state from the Ill state to be greater than that of from the $Well$ state. If however the probability of moving to the dead state is increased after being in the Ill state but less than in the Ill state a $Well\, after\, Ill$ state can be added to the model. The Ill state is then called a tunnel state.

We discuss two other situations in more detail below where the Markov processes described in this dissertation are not valid and for which the possible solution is more complicated than that described in the paragraph above.

5.3.1 Hidden Markov models

Hidden Markov models can be used when the true state of a process is not observed directly but another variables are observed which provides information on the true state of the process. This includes the case where the state of the system is observed with some error caused for example by misclassification of a measurement procedure[12]. Consider for example a sequence of measurements, $\{Y_t\}$, which are related to the state of a Markov process $\{X_t\}$ through the function $P[Y|X]$. $\{X_t\}$ is therefore the hidden Markov process. Given the state of the hidden process at time t, it is assumed an observation Y_t is independent of the state of the hidden process prior to t. We therefore have

$$P[Y_t|X_t, X_{t-1}, \dots, X_0] = P[Y_t|X_t] = f(y|x).$$

The goal is now to find the parameters of the hidden process that will maximise likelihood of observing an outcome $Y = \{Y_0, Y_1, \dots, Y_T\}$, i.e.

$$P[Y] = \sum_{W_0} P[Y|X]P[X]$$

where W_0 represents the state space of $\{X_t\}$. Bureau et al.[12] provides a more thorough discussion of Hidden Markov processes and the estimation of the parameters of the process with the inclusion of covariate effects.

The *msm* package has the ability to estimate the parameters of hidden Markov models and has been used by, amongst others, Mekonnen et al.[73]. Mekonnen et al. investigate the effect of lower CD4 counts on the life expectancy in a population of Ethiopian factory workers and used a hidden Markov model to account for the misspecification of CD4 counts.

Hidden Markov models are also used frequently in speech and optical recognition as indicated in the literature review in Section 1.3.

5.3.2 Dynamical systems models

The methods discussed in the first four chapters of this dissertation concern what is also known as cohort model. Such models consider a fixed amount of patients and model these patients over a period of time, typically until they are dead. The interaction between patients is ignored and the possible increase or decrease in the incidence of a disease is ignored. For infectious diseases the interaction between patients does however make a difference and the incidence of infectious disease can be seasonal. Consider for example influenza which typically occur between autumn and spring resulting in fluctuating incidence rates. One could use a cohort model to estimate the effect of some medical intervention, like vaccination, that reduces the severity of influenza on the life of patients that get the disease. Such a cohort model will however fail in including the possible effect of a reduced spread of the disease cause by vaccinated patients being less infective. Disease models which include patients interaction and where there is a constant in and outflow of patients are called dynamic models. Examples of such models are SIR type models which consider each patient to be susceptible, infectious or resistant; thus the name SIR models. These models are also a type of compartmental model, but allows for more flexibility in stating the rate at which patients move between states and does not necessarily satisfy the Markov assumption. SIR models are typically modelled as a system of partial differential equations of which the steady state distribution of the number of patients in each of the three compartments in the model is of interest. These models are also used to calculate what is called the basic reproduction number, written R_0, which represents the expected number of secondary people infected with a disease when one infective person is introduced into a population where everyone is susceptible. The disease is said to be endemic if $R_0 > 1$.

Hethcote[45] provides an overview of some of the methods involved in the modelling of infectious diseases. The “classic endemic model” discussed by Hethcote will be overviewed briefly to give an example of a SIR model and the interested reader is referred to the article by Hethcote for more details. We will consider a system where people can enter the susceptible state by being born, exit the susceptible state by moving to the infectious state or by dying. People therefore enter the infectious state by becoming infective from the susceptible state and can move to the resistant state when their infectious period ends or they exit the system when they die. Patients enter the resistant state from the infectious state when their infectious period ends and they stay in this state until they die. The rate of births is denoted by μ and the mortality rate is assumed to be equal to the birth rate so that the number of people in the population stay constant. The mortality rate is therefore also μ. The rate at which people move from the susceptible state to the infectious state is dependent on the contact rate between infectious and susceptible. The contact rate of people is denoted by β and it is assumed that one contact between an infectious and susceptible person causes the susceptible person to become infectious. The parameter β can be rescaled if the probability that such a contact does causes a susceptible person to become infectious is less than one. The duration of infectiousness is denoted by $\frac{1}{\gamma}$ so that people move from the infectious state to the resistant state at a rate γ. The system of differential equations is shown in Equation (103).

$$\begin{aligned}
\frac{dS}{dt} &= \mu N - \beta I \frac{S}{N} - \mu S \\
\frac{dI}{dt} &= \beta I \frac{S}{N} - \gamma I - \mu S \\
\frac{dR}{dt} &= \gamma I - \mu R \\
S(0) &= S_0 \geq 0 \\
I(0) &= I_0 \geq 0 \\
R(0) &= R_0 \geq 0
\end{aligned} \tag{103}$$

$$N(t) = S(t) + I(t) + R(t)$$

The aim is now to find the solutions to the system in Equation (103), i.e. $S(t), I(t), R(t)$. The system can be simplified by considering the proportion of people that are susceptible, infectious and resistant, $s(t) = \frac{S(t)}{N}, i(t) = \frac{I(t)}{N}, r(t) = \frac{R(t)}{N}$ and then only solving for two of these quantities in a new systems of differential equations since $s(t) + i(t) + r(t) = 1$. Hethcote indicates that the basic reproduction for the system in Equation (103) becomes

$$R_0 = \frac{\beta}{\gamma + \mu}.$$

5.4 Summary

We have discussed methods that can be used in disease modelling based on assumptions regarding the data and for diseases where patient interaction does not influence the effect of the disease and treatment thereof on society. These methods can be used when one is interested in estimating the morbidity, mortality and financial effect of diseases and different treatment options.

The methods are discussed based on longitudinal data where patients are followed for a period of time and where the state of each patient is known at certain time points. Furthermore we assume the state of each patient is known with complete certainty at each observation. The methods were discussed under the assumption of stationary Markov processes with suggestions being made on the treatment of nonstationary processes.

The methods were discussed and implemented for various observation schemes and it was indicated that Markov process can provide good estimates of the percentage of patients expected in each of the disease states. This information can be used if the cost and quality of life of being in such a state is known to estimate the total cost of patients, the quality of life of patients and the life expectancy of patients. The methods discussed can estimate the effect of covariates on process parameters, but we indicated that such estimates are not always estimated accurately unless state transitions are observed exactly.

Suggestions were made for the treatment of aggregate data, diseases where the state of patients are not known with complete certainty at every observation and diseases where patient interaction plays a role.

Appendix

1. Maximum likelihood estimation of a multinomial distribution

The proof of the ML estimator for a multinomial distribution is based on the proof found on page 21 of the textbook *Categorical Data Analysis* by Argest A.

Consider a multinomial experiment of n trials, each of which has $\mathscr{s}$ possible outcomes with probabilities $p_1, p_2, \dots, p_{\mathscr{s}}$.

If n observations from a multinomial experiment is available such that x_i is the number of times that the i^{th} possible outcome realised, the likelihood function of this process is

$$L(p|x) = \frac{n!}{x_1!, x_2!, \dots, x_{\mathscr{s}}!} {p_1}^{x_1} {p_2}^{x_2} \dots {p_{S-1}}^{x_{\mathscr{s}-1}} (1 - \sum_{i=1}^{\mathscr{s}-1} p_i)^{n - \sum_{i=1}^{\mathscr{s}-1} x_i} . \tag{104}$$

Taking the natural log of Equation (104) and omitting the terms without probabilities we get

$$\mathcal{L}(p|x) = x_1\, ln(p_1) + \dots + x_{\mathscr{s}-1}\, ln(p_{\mathscr{s}-1}) + \left(n - \sum_{i=1}^{\mathscr{s}-1} x_i\right) ln\, (1 - \sum_{i=1}^{\mathscr{s}-1} p_i). \tag{105}$$

Now, to find the ML estimate of p_j, we differentiate Equation (105) with respect to p_j and set equal to 0 to obtain

$$\frac{d\mathcal{L}(p|x)}{dp_j} = \frac{x_j}{\widehat{p_j}} - \frac{(n - \sum_{i=1}^{\mathscr{s}-1} x_i)}{1 - \sum_{i=1}^{\mathscr{s}-1} \widehat{p_i}} = 0$$

$$\Rightarrow x_j \left(1 - \sum_{i=1}^{\mathscr{s}-1} \widehat{p_i}\right) = \left(n - \sum_{i=1}^{\mathscr{s}-1} x_i\right) \widehat{p_j}$$

$$\Rightarrow x_j(\widehat{p_{\mathscr{s}}}) = (x_{\mathscr{s}})\widehat{p_j}. \tag{106}$$

Now sum both sides of the equality over all values of j and remembering that $\sum_{j=1}^{\mathscr{s}} \widehat{p_j} = 1$, we get

$$\sum_{j=1}^{\mathscr{s}} x_j(\widehat{p_S}) = \sum_{j=1}^{\mathscr{s}} (x_S)\widehat{p_j}$$

$$\Longrightarrow n(\widehat{p_s}) = x_s$$

$$\Longrightarrow \widehat{p_s} = \frac{x_s}{n}. \qquad (107)$$

Substituting Equation (107) into Equation (106) we get

$$\Rightarrow x_j \frac{x_s}{n} = (x_s)\widehat{p_J}$$

$$\Longrightarrow \widehat{p_J} = \frac{x_j}{n}.$$

This concludes the proof.

2. R code

```
rm(list=ls(all=TRUE))
mPower <- function(x, n) {
        power <- x
        if (n > 1) {
                for (i in 2:n) {
                        power <- power%*%x
                }
        }
        return(power)
}

pMatrix <- function(Q,nlim=30,t) {
        P <- matrix(0,nrow=nrow(Q),ncol=ncol(Q))
        diag(P) <- 1
        for (n in 1:nlim) {
                sum <- mPower(Q,n)*(t^n)/factorial(n)
                P <- P + sum
        }
        return(P)
}

QTime <- rbind(c(-3.6,0.8,1.3,0.4,0.9,0.2),c(1.2,-3.2,0.3,0.2,1.4,0.1),c(0.3,1.4,-3.1,0.7,0.5,0.2),c(0.3,0.5,1.2,-3.5,0.4,1.1),c(0.1,1.2,0.7,0.5,-2.9,0.4),c(0,0,0,0,0,0))
apply(QTime,1,sum)

tpm <- proc.time()
pMatrix(QTime,nlim=10,t=1)
proc.time() - tpm

tpm <- proc.time()
pMatrix(QTime,nlim=20,t=1)
proc.time() - tpm

tpm <- proc.time()
pMatrix(QTime,nlim=30,t=1)
proc.time() - tpm

tpm <- proc.time()
pMatrix(QTime,nlim=40,t=1)
proc.time() - tpm

tpm <- proc.time()
eigen(QTime)$vector%*%diag(exp(eigen(QTime)$value*1))%*%solve(eigen(QTime)$vector)
proc.time() - tpm

rm(QTime)
rm(tpm)

#####Simulate Markov chain#######

RecordLongData <- function(TPM,AbState,InitialState,k,T,TimeSteps=1,saad) {
        States <- 1:ncol(TPM)
        set.seed(saad)
```

```
    nStates <- ncol(TPM)
    Obs <- c(1,InitialState,0); #Obs: PatID|State|Time at which Pat go in

    for (PatID in 1:k) {
        if (PatID > 1) Obs <- rbind(Obs,c(PatID,InitialState,0))
        CurrentState <- InitialState
        for (time in seq(0+TimeSteps,T,TimeSteps)) {
            if (CurrentState != AbState) {
                Probs <- TPM[CurrentState ,]
                Pos <- rmultinom(1,1,Probs)
                NewState <- States[Pos==1]
                Obs = rbind(Obs,c(PatID,NewState,time))
            CurrentState <- NewState
            }
        }
    }
    return(Obs)
}

######Function for nonstationary chain######

RecordLongDataNonStat <- function(TPM,AbState,InitialState,k,T,TimeSteps=1,saad) {
    set.seed(saad)
    nStates <- ncol(TPM)
    States <- 1:nStates
    Obs <- c(1,InitialState,0); #Obs: PatID|State|Time at which Pat go in
    for (PatID in 1:k) {
        if (PatID > 1) Obs <- rbind(Obs,c(PatID,InitialState,0))
        CurrentState <- InitialState
        for (time in seq(TimeSteps,T,TimeSteps)) {
                TPM[1,1] <- exp(-( (0.2*time)^2.5+(0.2*(time-TimeSteps))^2.5 ))
                Rem <- 1-TPM[1,1]
                TPM[1,2:(AbState-1)] <- rep(Rem*0.8/(nStates-2),nStates-2)
                TPM[1,AbState] <- 1-sum(TPM[1,-AbState])
                if (CurrentState != AbState) {
                    Probs <- TPM[CurrentState ,]
                    Pos <- rmultinom(1,1,Probs)
                    NewState <- States[Pos==1]
                    Obs = rbind(Obs,c(PatID,NewState,time))
                    CurrentState <- NewState
                }
        }
    }
    return(Obs)
}

############Jump process###############"

RecordJumpProcess <- function(Q,k,InState,AbState,T,saad) {
    States <- 1:ncol(Q)
    set.seed(saad)
    for (Pat in 1:k) {
        CurrentState <- InState
        CumTime <- 0
        PatData <- c(Pat,CurrentState,0)
        sentOuter <- 1
        while(sentOuter == 1) {
            timeJump <- rexp(1,rate=-Q[CurrentState,CurrentState])
            CumTime <- CumTime + timeJump
            if (CumTime < T) {
                NewStateProbs <- Q[CurrentState,]/(-Q[CurrentState,CurrentState])
                NewStateProbs[CurrentState] <- 0
                Pos <- rmultinom(1,1,NewStateProbs)
                NewState <- States[Pos==1]
                PatData <- rbind(PatData,c(Pat,NewState,CumTime))
                CurrentState <- NewState
                if (CurrentState == AbState) sentOuter <- 0
            }
            if (CumTime >= T) {
                PatData <- rbind(PatData,c(Pat,CurrentState,T))
                sentOuter <- 0
            }
        }
        if (Pat == 1) Obs <- PatData
        else Obs <- rbind(Obs,PatData)
    }
    return(Obs)
}

#####Include effect of gender on transition rate#####

RecordJumpProcessGenderEffect <- function(Q,GenderVector,GenderEffect,k,InState,AbState,T,saad) {
    States <- 1:ncol(Q)
    B012 <- log(Q[1,2])
    set.seed(saad)
    for (Pat in 1:k) {
        Q[1,2] <- exp(B012+GenderEffect*GenderVector[Pat])
        Q[1,1] <- -sum(Q[1,-1])
        CurrentState <- InState
        CumTime <- 0
        PatData <- c(Pat,CurrentState,0)
        sentOuter <- 1
        while(sentOuter == 1) {
            timeJump <- rexp(1,rate=-Q[CurrentState,CurrentState])
            CumTime <- CumTime + timeJump
            if (CumTime < T) {
                NewStateProbs <- Q[CurrentState,]/(-Q[CurrentState,CurrentState])
                NewStateProbs[CurrentState] <- 0
                Pos <- rmultinom(1,1,NewStateProbs)
                NewState <- States[Pos==1]
                PatData <- rbind(PatData,c(Pat,NewState,CumTime))
                CurrentState <- NewState
                if (CurrentState == AbState) sentOuter <- 0
            }
            if (CumTime >= T) {
                PatData <- rbind(PatData,c(Pat,CurrentState,T))
                sentOuter <- 0
            }
        }
        if (Pat == 1) Obs <- PatData
        else Obs <- rbind(Obs,PatData)
    }
    return(Obs)
}

#####Include effect of age and gender  on transition rate#####

RecordJumpProcessGenderAgeEffect <-
function(Q,GenderVector,GenderEffect,AgeVector,AgeEffect,k,InState,AbState,T,saad) {
    States <- 1:nStates
    B012 <- log(Q[1,2])
    set.seed(saad)
    for (Pat in 1:k) {
        Q[1,2] <- exp(B012+GenderEffect*GenderVector[Pat]+AgeEffect*AgeVector[Pat])
        Q[1,1] <- -sum(Q[1,-1])
        CurrentState <- InState
        CumTime <- 0
        PatData <- c(Pat,CurrentState,0)
        sentOuter <- 1
        while(sentOuter == 1) {
            timeJump <- rexp(1,rate=-Q[CurrentState,CurrentState])
            CumTime <- CumTime + timeJump
            if (CumTime < T) {
                NewStateProbs <- Q[CurrentState,]/(-Q[CurrentState,CurrentState])
                NewStateProbs[CurrentState] <- 0
                Pos <- rmultinom(1,1,NewStateProbs)
                NewState <- States[Pos==1]
                PatData <- rbind(PatData,c(Pat,NewState,CumTime))
                CurrentState <- NewState
                if (CurrentState == AbState) sentOuter <- 0
            }
            if (CumTime >= T) {
                PatData <- rbind(PatData,c(Pat,CurrentState,T))
                sentOuter <- 0
            }
        }
        if (Pat == 1) Obs <- PatData
        else Obs <- rbind(Obs,PatData)
    }
    return(Obs)
}

#####################"Stationary Markov jump procees######################

###Examples 1 and 2

QSim <- rbind(c(-0.35,0.15,0.2),c(0.2,-0.4,0.2),c(0,0,0)); #Example 1
QSim <- rbind(c(-1.13,0.2,0.8,0.03,0.1),c(0.8,-1.11,0.2,0.01,0.1),c(0.4,0.9,-1.42,0.02,0.1),c(1.3,0.8,0.5,-
2.68,0.08),c(0,0,0,0,0)); #Example 2

DeathState <- ncol(QSim)
nStates <- ncol(QSim)
NumPatients <- 100
MaxTime <- 5
InState=1

TPM <- pMatrix(Q,nlim=30,t=1/999999999999)
eye <- matrix(0,nrow=2,ncol=2)
```

```
diag(eye) <- 1
sum(solve(eye-TPM[1:2,1:2])[1,])/999999999999

JumpData <-
RecordJumpProcess(Q=QSim,k=NumPatients,InState=1,AbState=DeathState,T=MaxTime,saad=1010)
table(JumpData[,2])

NSample <- matrix(ncol=ncol(QSim),nrow=nrow(QSim),0)
ASample <- rep(0,nStates)

for (rcount in 2:nrow(JumpData) ) {
        if ( JumpData[rcount,1] == JumpData[rcount-1,1] )  {
                NSample[JumpData[rcount-1,2],JumpData[rcount,2]] <- NSample[JumpData[rcount-
1,2],JumpData[rcount,2]] + 1
                ASample[JumpData[rcount-1,2]] <- ASample[JumpData[rcount-1,2]] + (JumpData[rcount,3]-
JumpData[rcount-1,3])
        }
        if ( JumpData[rcount,1] == (JumpData[rcount-1,1] + 1) )  {
                ASample[JumpData[rcount-1,2]] <- ASample[JumpData[rcount-1,2]] + (MaxTime-
JumpData[rcount-1,3])
        }
}
ASample[DeathState] <- ASample[DeathState] + (MaxTime - JumpData[nrow(JumpData),3])
sum(ASample)

sum(ASample[-ncol(QSim)])/NumPatients

QHat <- matrix(ncol=ncol(QSim),nrow=ncol(QSim))

for (i in 1:ncol(NSample)) {
        for (j in 1:ncol(NSample)) {
                if (i != j) QHat[i,j] <- NSample[i,j]/ASample[i]
        }
        QHat[i,i] <- -apply(QHat,1,sum,na.rm=TRUE)[i]
}
QSim
QHat

sum((QHat - QSim)^2)

#Estimates from msm
library(msm)

ObsDF <- data.frame(JumpData)
names(ObsDF) <- c("PatID","State","Time")
#statetable.msm(State,PatID,data=ObsDF)
FirstQ <- rbind(c(-1.4,0.2,0.8),c(0.8,-1.2,0.2),c(0,0,0))
#FirstQ <- rbind(c(-1.4,0.2,0.8,0.3,0.1),c(0.8,-1.2,0.2,0.1,0.1),c(0.4,0.9,-1.6,0.2,0.1),c(0.2,0.4,0.3,-
1.7,0.8),c(0,0,0,0,0))

Qmsm <- crudeinits.msm(State~Time,PatID,data=ObsDF,qmatrix=FirstQ )
Q.msm <- msm(State~Time,PatID,data=ObsDF,qmatrix=FirstQ,exacttimes=TRUE )

totlos.msm(Q.msm)
```

```
#What if we observed the data only every time point and treated it as discrete data?

#Extract discrete time point observations

for (Pat in 1:NumPatients) {
        PatData <- JumpData[JumpData[,1]==Pat,]
                PatDiscr <- c(Pat,InState,0)
                for (time in 1:MaxTime) {
                        Pos <- PatData[PatData[,3]<=time,]
                        if (length(Pos) == 3) LastState <- c(Pat,Pos[2],time)
                        if (length(Pos) > 3 ) LastState <- Pos[nrow(Pos),]
                        #if (length(PatDiscr) == 3) testRow <- PatDiscr
                        #if (length(PatDiscr) > 3) testRow <- PatDiscr[nrow(PatDiscr),]
                        #if ( sum(testRow - LastState) != 0 ) PatDiscr <-
rbind(PatDiscr,c(Pat,LastState[2],time))
                        PatDiscr <- rbind(PatDiscr,c(Pat,LastState[2],time))
                }
        if (Pat == 1) DiscrObs <- PatDiscr
        if (Pat > 1) DiscrObs <- rbind(DiscrObs,PatDiscr)
}

#Fit TPM

TCM <- matrix(ncol=ncol(QSim),nrow=nrow(QSim),0)

for (rcount in 2:nrow(DiscrObs) ) {
        if ( DiscrObs[rcount,1] == DiscrObs[rcount-1,1] )  {
                TCM[DiscrObs[rcount-1,2],DiscrObs[rcount,2]] <- TCM[DiscrObs[rcount-
1,2],DiscrObs[rcount,2]] + 1
        }
}

TPMHat <- matrix(0,nrow=nrow(QSim),ncol=nrow(QSim))

for (i in 1:nrow(QSim)) {
        for (j in 1:nrow(QSim)) {
                TPMHat[i,j] <- TCM[i,j]/apply(TCM,1,sum)[i]
        }
}
DiscrTPMHat <- TPMHat
ExactTPMHat <- pMatrix(QHat,t=1)
TrueTPMHat <- pMatrix(QSim,t=1)
#The TPM of QHat is closer to the TPM of Q than TPMHat

#Calculate true state prevalence and plot estimated prevalence on that

maxTime <- 30
TrueDistr <- t(as.matrix(rep(0,ncol(TPMHat))))
TrueDistr[InState] <- 1

DiscreteDistr <- TrueDistr
ExactDistr <- TrueDistr

PrevMatrix <- matrix(0,nrow=ncol(TPMHat),ncol=5)
PrevMatrix[,1] <- 1:5
PrevMatrix[,3] <- TrueDistr
```

```
PrevMatrix[,4] <- DiscreteDistr
PrevMatrix[,5] <- ExactDistr

for (time in 1:maxTime) {
        TrueDistr <- TrueDistr%*%TrueTPMHat
        DiscreteDistr <- DiscreteDistr%*%DiscrTPMHat
        ExactDistr <- ExactDistr%*%ExactTPMHat

        PrevMatrix <-
rbind(PrevMatrix,cbind(1:5,rep(time,ncol(TPMHat)),t(TrueDistr),t(DiscreteDistr),t(ExactDistr)))
}

PrevMatrix <- data.frame(PrevMatrix)
names(PrevMatrix) <- c("State","Time","TruePrev","DiscrPrev","ExactPrev")

State1 <- PrevMatrix[PrevMatrix[,1]==1,2:5]
State2 <- PrevMatrix[PrevMatrix[,1]==2,2:5]
State3 <- PrevMatrix[PrevMatrix[,1]==3,2:5]
State4 <- PrevMatrix[PrevMatrix[,1]==4,2:5]
State5 <- PrevMatrix[PrevMatrix[,1]==5,2:5]

par(mfrow=c(3,2))

plot(State1$Time,State1$TruePrev,type="l",col="black",xlab="Time",ylab="Prevalence",main="State1")
lines(State1$Time,State1$DiscrPrev,col="blue")
lines(State1$Time,State1$ExactPrev,col="green")

plot(State2$Time,State2$TruePrev,type="l",col="black",xlab="Time",ylab="Prevalence",main="State2")
lines(State2$Time,State2$DiscrPrev,col="blue")
lines(State2$Time,State2$ExactPrev,col="green")

plot(State3$Time,State3$TruePrev,type="l",col="black",xlab="Time",ylab="Prevalence",main="State3",ylim=c(
0,0.3))
lines(State3$Time,State3$DiscrPrev,col="blue")
lines(State3$Time,State3$ExactPrev,col="green")

plot(State4$Time,State4$TruePrev,type="l",col="black",xlab="Time",ylab="Prevalence",main="State4",ylim=c(
0,0.015))
lines(State4$Time,State4$DiscrPrev,col="blue")
lines(State4$Time,State4$ExactPrev,col="green")

plot(State5$Time,State5$TruePrev,type="l",col="black",xlab="Time",ylab="Prevalence",main="State5")
lines(State5$Time,State5$DiscrPrev,col="blue")
lines(State5$Time,State5$ExactPrev,col="green")

#Maximise the likelihood function with the nlm method

likeNoCov <- function(Beta) {
        logl <- 0
        BetaCount <- 1
        for (j in 1:nStates) {
                if (j != DeathState) {
                        sum2 <- 0
                        for (k in 1:nStates) {
                                if (j != k ) {
                                        logl <- logl + NSample[j,k]*log(Beta[BetaCount])
```

```
                    sum2 <- sum2 + Beta[BetaCount]
                    BetaCount <- BetaCount + 1
                }
            }
            logl <- logl - ASample[j]*sum2
        }
    }
    return(-logl)
}

As <- c(-0.205,0.02,0,0.2,-0.025,0,0.05,0.05,0)
As <- c(0.3,0.3,0.1,0.2)
As <- rep(1,4)
As <- c(0.2,0.15,0.15,0.25)
As <- c(0.2,0.8,0.3,0.1,0.8,0.2,0.1,0.1,0.4,0.9,0.2,0.1,0.2,0.4,0.3,0.8)
As <- c(0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5)

#Data <- rbind(NSample,ASample)
library(stats4)
nlm(likeNoCov,p=As)

#####################"Stationary Markov jump procees with gender effect#######################
#Example 3

QSim <- rbind(c(-0.25,0.1,0.15),c(0.01,-0.16,0.15),c(0,0,0))
QSim <- rbind(c(-2.5,2,0.5),c(1.5,-2,0.5),c(0,0,0))
QSim <- rbind(c(-1.4,0.2,0.8,0.3,0.1),c(0.8,-1.2,0.2,0.1,0.1),c(0.4,0.9,-1.6,0.2,0.1),c(0.2,0.4,0.3,-
1.7,0.8),c(0,0,0,0,0))
QSim <- rbind(c(-0.35,0.15,0.2),c(0.2,-0.4,0.2),c(0,0,0))

#Generate gender
NumPatients <- 1000
set.seed(1604)
Gender <- rbinom(n=NumPatients,size=1,prob=0.7)
GenderEffect <- 0.4
exp(GenderEffect)
#exp(Gender*GenderEffect)

DeathState <- ncol(QSim)
nStates <- ncol(QSim)

NumPatients <- 1000
MaxTime <- 50

JumpData <-
RecordJumpProcessGenderEffect(Q=QSim,GenderVector=Gender,GenderEffect=GenderEffect,AbState=DeathS
tate,InState=1,k=NumPatients,T=MaxTime,saad=1010)

NSample <- vector("list",NumPatients)
ASample <- vector("list",NumPatients)

for (Pat in 1:NumPatients ) {
    NSample[[Pat]] <- matrix(0,nrow=nStates,ncol=nStates)
    ASample[[Pat]] <- matrix(0,nrow=nStates,ncol=1)

    PatData <- JumpData[JumpData[,1] == Pat,]
```

```
        for (rcount in 2:nrow(PatData) ) {
                NSample[[Pat]][PatData[rcount-1,2],PatData[rcount,2]] <- NSample[[Pat]][PatData[rcount-
1,2],PatData[rcount,2]] + 1
                ASample[[Pat]][PatData[rcount-1,2]] <- ASample[[Pat]][PatData[rcount-1,2]] +
(PatData[rcount,3]-PatData[rcount-1,3])
        }
        ASample[[Pat]][PatData[nrow(PatData),2]] <- ASample[[Pat]][PatData[nrow(PatData),2]] + (MaxTime -
PatData[nrow(PatData),3])

}

testSumN <- matrix(0,nrow=nStates,ncol=nStates)
testSumA <- matrix(0,nrow=nStates,ncol=1)

for (i in 1:NumPatients) {
        testSumN <- testSumN + NSample[[i]]
        testSumA <- testSumA + ASample[[i]]
}
sum(testSumA)

QHat <- matrix(ncol=ncol(QSim),nrow=ncol(QSim))

for (i in 1:ncol(testSumN)) {
        for (j in 1:ncol(testSumN )) {
                if (i != j) QHat[i,j] <- testSumN[i,j]/testSumA[i]
        }
        QHat[i,i] <- -apply(QHat,1,sum,na.rm=TRUE)[i]
}
QSim
QHat

#Gender = 1 is in NSample[[1]] and Gender = 0 is in NSample[[2]]. ASample is similar

likeGender <- function(Beta) {
        logl <- 0
        for (PatCount in 1:NumPatients) {
                BetaCount <- 2
                for (j in 1:nStates) {
                        if (j != DeathState) {
                                sum2 <- 0
                                for (k in 1:nStates) {
                                        if (j == 1 & k == 2) {
                                                logl <- logl +
NSample[[PatCount]][1,2]*(Beta[1]+Beta[length(Beta)]*Gender[PatCount])
                                                sum2 <- sum2 +
exp(Beta[1]+Beta[length(Beta)]*Gender[PatCount])
                                        } else if (j != k ) {
                                                logl <- logl + NSample[[PatCount]][j,k]*Beta[BetaCount]
                                                sum2 <- sum2 + exp(Beta[BetaCount])
                                                BetaCount <- BetaCount + 1
                                        }
                                }
                                logl <- logl - ASample[[PatCount]][j]*sum2
                        }
                }
        }
```

```
		return(-logl)
}

B121Guess <- -15
exp(B121Guess)

Betas <- c(log(c(0.2,0.15,0.15,0.2)),0.7)
Betas <- c(log(c(2,2,2,2)),1.5)
Betas <- c(log(QHat[1,2])-B121Guess*mean(Gender),log(c(QHat[1,3],QHat[2,1],QHat[2,3])),B121Guess)

Betas <- c(rep(log(0.5),4),1.2)

As <- c(0.2,0.8,0.3,0.1,0.8,0.2,0.1,0.1,0.4,0.9,0.2,0.1,0.2,0.4,0.3,0.8)
As <- c(0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5)

Betas <- c(rep(log(0.5),16),1.2)
Betas <- c(log(c(0.4,1,0.5,0.5,0.2,0.4,0.1,0.3,0.5,1.4,0.1,0.4,0.6,0.2,0.5,0.2)),0.4)
Betas <- c(log(c(0.2,0.8,0.3,0.1,0.8,0.2,0.1,0.1,0.4,0.9,0.2,0.1,0.2,0.4,0.3,0.8)),0.4)

library(stats4)

test <- nlm(likeGender,p=Betas)
 exp(test$estimate)

QMale <- rbind(c(-exp(test$estimate)[1]-
exp(test$estimate)[2],exp(test$estimate)[1],exp(test$estimate)[2]),c(exp(test$estimate)[3],-
exp(test$estimate)[3]-exp(test$estimate)[4],exp(test$estimate)[4]),c(0,0,0))
QFemale <- rbind(c(-exp(test$estimate[1]+test$estimate[5])-
exp(test$estimate)[2],exp(test$estimate[1]+test$estimate[5]),exp(test$estimate)[2]),c(exp(test$estimate)[3],-
exp(test$estimate)[3]-exp(test$estimate)[4],exp(test$estimate)[4]),c(0,0,0))

QMale
QFemale

#What does msm give?

library(msm)

ObsDF <- data.frame(JumpData)
names(ObsDF) <- c("PatID","State","Time")
ObsDF$Sex <- Gender[ObsDF[,1]]

#statetable.msm(State,PatID,data=ObsDF)
FirstQ <- rbind(c(-1.5,0.2,0.8),c(0.8,-1.4,0.2),c(0,0,0))

FirstQ <- rbind(c(-1.5,0.2,0.8,0.3,0.5),c(0.8,-1.4,0.2,0.3,0.1),c(0.8,0.1,-1.3,0.3,0.1),c(0.8,0.2,0.3,-
1.4,0.1),c(0,0,0,0,0))
QmsmIn <- crudeinits.msm(State~Time,PatID,data=ObsDF,qmatrix=FirstQ )

Qmsm <-
msm(State~Time,PatID,data=ObsDF,qmatrix=QmsmIn,exacttimes=TRUE,opt.method="nlm",covariates=~Sex)
qmatrix.msm(Qmsm,covariates=list(Sex=1))

###############"Stationary Markov jump procees with gender and age effect##################
#Example 4
```

```
QSim <- rbind(c(-0.25,0.1,0.15),c(0.01,-0.16,0.15),c(0,0,0))
QSim <- rbind(c(-2.5,2,0.5),c(1.5,-2,0.5),c(0,0,0))
QSim <- rbind(c(-1.4,0.2,0.8,0.3,0.1),c(0.8,-1.2,0.2,0.1,0.1),c(0.4,0.9,-1.6,0.2,0.1),c(0.2,0.4,0.3,-
1.7,0.8),c(0,0,0,0,0))
QSim <- rbind(c(-0.35,0.15,0.2),c(0.2,-0.4,0.2),c(0,0,0))

NumPatients <- 1000
MaxTime <- 50

#Generate gender

set.seed(1604)
Gender <- rbinom(n=NumPatients,size=1,prob=0.5)
GenderEffect <- 0.4
exp(GenderEffect)
#exp(Gender*GenderEffect)

#Generate age
set.seed(1604)
Age <- runif(n=NumPatients,min=0,max=100)
AgeEffect <- 0.05
exp(AgeEffect)
#exp(Age*AgeEffect)

AgePlot <- Age[order(Age)]

plot(AgePlot,exp(QSim[1,2]+AgePlot*AgeEffect+GenderEffect),type='l',col='green',ylab='Transition intensity
from state 1 to 2',xlab='Age')
lines(AgePlot,exp(QSim[1,2]+AgePlot*AgeEffect),col='blue')
legend(x=50,y=250,c('Female','Male'),col=c('green','blue'),pch=18)

DeathState <- ncol(QSim)
nStates <- ncol(QSim)

JumpData <-
RecordJumpProcessGenderAgeEffect(Q=QSim,GenderVector=Gender,GenderEffect=GenderEffect,AgeVector=
Age,AgeEffect=AgeEffect,AbState=DeathState,InState=1,k=NumPatients,T=MaxTime,saad=1010)

NSample <- vector("list",NumPatients)
ASample <- vector("list",NumPatients)

for (Pat in 1:NumPatients ) {
        NSample[[Pat]] <- matrix(0,nrow=nStates,ncol=nStates)
        ASample[[Pat]] <- matrix(0,nrow=nStates,ncol=1)
        PatData <- JumpData[JumpData[,1] == Pat,]
        for (rcount in 2:nrow(PatData) ) {
                NSample[[Pat]][PatData[rcount-1,2],PatData[rcount,2]] <- NSample[[Pat]][PatData[rcount-
1,2],PatData[rcount,2]] + 1
                ASample[[Pat]][PatData[rcount-1,2]] <- ASample[[Pat]][PatData[rcount-1,2]] +
(PatData[rcount,3]-PatData[rcount-1,3])
        }
        ASample[[Pat]][PatData[nrow(PatData),2]] <- ASample[[Pat]][PatData[nrow(PatData),2]] + (MaxTime -
PatData[nrow(PatData),3])

}
```

```
testSumN <- matrix(0,nrow=nStates,ncol=nStates)
testSumA <- matrix(0,nrow=nStates,ncol=1)

for (i in 1:NumPatients) {
    testSumN <- testSumN + NSample[[i]]
    testSumA <- testSumA + ASample[[i]]
}
sum(testSumA)

QHat <- matrix(ncol=ncol(QSim),nrow=ncol(QSim))

for (i in 1:ncol(testSumN)) {
    for (j in 1:ncol(testSumN )) {
        if (i != j) QHat[i,j] <- testSumN[i,j]/testSumA[i]
    }
    QHat[i,i] <- -apply(QHat,1,sum,na.rm=TRUE)[i]
}
QSim
QHat

likeGenderAge <- function(Beta) {
    logl <- 0
    for (PatCount in 1:NumPatients) {
        BetaCount <- 2
        for (j in 1:nStates) {
            if (j != DeathState) {
                sum2 <- 0
                for (k in 1:nStates) {
                    if (j == 1 & k == 2) {
                        logl <- logl +
NSample[[PatCount]][1,2]*(Beta[1]+Beta[length(Beta)-
1]*Gender[PatCount]+Beta[length(Beta)]*Age[PatCount])
                        sum2 <- sum2 + exp(Beta[1]+Beta[length(Beta)-
1]*Gender[PatCount]+Beta[length(Beta)]*Age[PatCount])
                    } else if (j != k ) {
                        logl <- logl + NSample[[PatCount]][j,k]*Beta[BetaCount]
                        sum2 <- sum2 + exp(Beta[BetaCount])
                        BetaCount <- BetaCount + 1
                    }
                }
                logl <- logl - ASample[[PatCount]][j]*sum2
            }
        }
    }
    return(-logl)
}

As <- c(0.2,0.8,0.3,0.1,0.8,0.2,0.1,0.1,0.4,0.9,0.2,0.1,0.2,0.4,0.3,0.8)
As <- c(0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5)
Betas <- c(log(c(0.2,0.15,0.15,0.2)),0.7,0.1)
Betas <- c(log(c(2,2,2,2)),1.5,1)
B121Guess <- 0.5337752
B122Guess <- 0.35
exp(B122Guess*100)
```

```
Betas <- c(log(QHat[1,2])-B121Guess*mean(Gender)-
B122Guess*mean(Age),log(c(QHat[1,3],QHat[2,1],QHat[2,3])),B121Guess,B122Guess)
nlm(likeGenderAge,p=Betas)$estimate

Betas <- c(rep(log(0.5),16),1.2,1.05)
Betas <- c(log(c(0.4,1,0.5,0.5,0.2,0.4,0.1,0.3,0.5,1.4,0.1,0.4,0.6,0.2,0.5,0.2)),0.4,0.05)
Betas <- c(log(c(0.2,0.8,0.3,0.1,0.8,0.2,0.1,0.1,0.4,0.9,0.2,0.1,0.2,0.4,0.3,0.8)),0.4,0.05)

library(stats4)

test <- nlm(likeGenderAge,p=Betas)
exp(test$estimate)

#What does msm give?

library(msm)
ObsDF <- data.frame(JumpData)
names(ObsDF) <- c("PatID","State","Time")
ObsDF$Sex <- Gender[ObsDF[,1]]
ObsDF$Age <- Age[ObsDF[,1]]

#statetable.msm(State,PatID,data=ObsDF)
#FirstQ <- rbind(c(-1.5,0.2,0.8,0.3,0.5),c(0.8,-1.4,0.2,0.3,0.1),c(0.8,0.1,-1.3,0.3,0.1),c(0.8,0.2,0.3,-
1.4,0.1),c(0,0,0,0,0))
FirstQ <- rbind(c(-1,0.2,0.8),c(0.8,-1,0.2),c(0,0,0))

QmsmIn <- crudeinits.msm(State~Time,PatID,data=ObsDF,qmatrix=FirstQ )
Qmsm <-
msm(State~Time,PatID,data=ObsDF,qmatrix=QmsmIn,exacttimes=TRUE,opt.method="nlm",covariates=~Sex+A
ge)
qmatrix.msm(Qmsm,covariates=list(Sex=0,Age=0))

###########################Stationary Markov Chain##########################################
#Example 5

TPM <-
rbind(c(0.7,0.1,0.05,0.05,0.1),c(0.13,0.2,0.35,0.2,0.12),c(0.1,0.05,0.1,0.5,0.25),c(0.05,0.05,0.1,0.4,0.4),c(0,0,0,0,
1))
TPM <-
rbind(c(0.7,0.1,0.05,0.05,0.1),c(0.1,0.3,0.35,0.05,0.2),c(0.1,0.05,0.5,0.05,0.3),c(0.05,0.05,0.1,0.4,0.4),c(0,0,0,0,1
))

apply(TPM,1,sum)
#How long does it take for the process to reach the death state if the process starts in state 1?
Que <- TPM[1:(nrow(TPM)-1),1:(nrow(TPM)-1)]
eye <- matrix(0,ncol=ncol(Que),nrow=nrow(Que)); diag(eye) <- 1
sum(solve((eye - Que))[1,])

InitialState=1
K=100
AbState <- 5
StatMarkovChain <- RecordLongData(TPM=TPM,AbState=AbState,InitialState=InitialState,k=K,T=3,saad=1010)

#Calculate the transition counts

TCM <- matrix(ncol=ncol(TPM),nrow=nrow(TPM),0)
```

```
for (rcount in 2:nrow(StatMarkovChain) ) {
        if ( StatMarkovChain[rcount,1] == StatMarkovChain[rcount-1,1] )  {
                TCM[StatMarkovChain[rcount-1,2],StatMarkovChain[rcount,2]] <-
TCM[StatMarkovChain[rcount-1,2],StatMarkovChain[rcount,2]] + 1
        }
}
TCM[nrow(TPM),ncol(TPM)] <- 1

#Make sure we get the same estimate for the FCM as that calculated by statetable.msm
library(msm)

ObsDF <- data.frame(StatMarkovChain)
names(ObsDF) <- c("PatID","State","Time")
statetable.msm(State,PatID,data=ObsDF)

TPMHat <- matrix(0,nrow=nrow(TPM),ncol=nrow(TPM))
for (i in 1:nrow(TPM)) {
        for (j in 1:nrow(TPM)) {
                TPMHat[i,j] <- TCM[i,j]/apply(TCM,1,sum)[i]
        }
}

TPM
round(TPMHat,3)
SSD <- sum((TPM-TPMHat)^2)
apply(TPMHat,1,sum)

#How long did it take the process to reach the absorbing state?

SurvTime <- StatMarkovChain[StatMarkovChain[,2] == AbState,3]
#Did everybidy die?
length(SurvTime)

summary(SurvTime)

#Make observed vs Expected plot

maxTime <- max(StatMarkovChain[,3])
ExpDistr <- t(as.matrix(rep(0,ncol(TPM))))
ExpDistr[InitialState] <- 1

for (time in 0:maxTime) {
        TimePrev <- data.frame(table(StatMarkovChain[StatMarkovChain[,3]==time,2]))
        TimePrev$Obs <- TimePrev[,2]/K
        TimePrev[,1] <- as.numeric(TimePrev[,1])
        for (stateCount in 1:ncol(TPM)) {
                if(is.na(match(stateCount ,TimePrev[,1]))) TimePrev <- rbind(TimePrev,c(stateCount,0,0))
        }
        TimePrev <- TimePrev[order(TimePrev[,1]),]
        TimePrev[AbState,3] <- 1-sum(TimePrev[TimePrev[,1] != AbState,3])

        TimePrev$Exp <- t(ExpDistr)
        ExpDistr <- ExpDistr%*%TPMHat
```

```
		if (time == 0) PrevMatrix <- cbind(TimePrev[,c(1,3,4)],rep(time,nrow(TimePrev)))
		if (time > 0) PrevMatrix <- rbind(PrevMatrix,cbind(TimePrev[,c(1,3,4)],rep(time,nrow(TimePrev))))
}
names(PrevMatrix) <- c("State","ObsPrev","ExpPrev","Time")
State1 <- PrevMatrix[PrevMatrix[,1]==1,2:4]
State2 <- PrevMatrix[PrevMatrix[,1]==2,2:4]
State3 <- PrevMatrix[PrevMatrix[,1]==3,2:4]
State4 <- PrevMatrix[PrevMatrix[,1]==4,2:4]
State5 <- PrevMatrix[PrevMatrix[,1]==5,2:4]

par(mfrow=c(3,2))

plot(State1$Time,State1$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State1")
lines(State1$Time,State1$ExpPrev,col="blue")

plot(State2$Time,State2$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State2")
lines(State2$Time,State2$ExpPrev,col="blue")

plot(State3$Time,State3$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State3")
lines(State3$Time,State3$ExpPrev,col="blue")

plot(State4$Time,State4$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State4")
lines(State4$Time,State4$ExpPrev,col="blue")

plot(State5$Time,State5$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State5")
lines(State5$Time,State5$ExpPrev,col="blue")

##########################Nonstationary Markov Chain########################################
#Example 6

#TPMSim <-
rbind(c(0.7,0.0,0.01,0.01,0.01),c(0,0.33,0.35,0.31,0.01),c(0,0,0.29,0.2,0.5,0.01),c(0,0.14,0.45,0.4,0.01),c(0,0,0,0,1
))
TPMSim <-
rbind(c(0.7,0.01,0.01,0.01,0.1),c(0.13,0.2,0.35,0.2,0.12),c(0.1,0.05,0.1,0.5,0.25),c(0.05,0.05,0.1,0.4,0.4),c(0,0,0,
0,1))

XTime <- 1:20
Hazard <- 0.2*2.5*XTime^1.5
plot(XTime,Hazard,type='l',xlab='Time',ylab='Hazard function')

InitialState=1
K <- 100
AbState <- 5
NonStatMarkovChain <-
RecordLongDataNonStat(TPM=TPMSim,AbState=AbState,InitialState=InitialState,k=K,T=3,saad=1010)

#Calculate the transition counts and probabilities

MaxTime <- max(NonStatMarkovChain[,3])
TCMList <- vector("list",MaxTime)
TPMList <- vector("list",MaxTime)

for (timeCount in 0:(MaxTime-1)) {
		TimeData <- NonStatMarkovChain[NonStatMarkovChain[,3]==timeCount |
NonStatMarkovChain[,3]==timeCount+1,]
```

```
        TCM <- matrix(ncol=ncol(TPMSim),nrow=nrow(TPMSim),0)
        for (rcount in 1:(nrow(TimeData)-1) ) {
                if ( TimeData[rcount,1] == TimeData[rcount+1,1] )  {
                        TCM[TimeData[rcount,2],TimeData[rcount+1,2]] <-
TCM[TimeData[rcount,2],TimeData[rcount+1,2]] + 1
                }
        }
        TCM[AbState,AbState] <- 1
        TPMEst <- TCM
        for (rows in 1:nrow(TCM)) {
                if (sum(TCM[rows,]) > 0) TPMEst[rows,] <- TCM[rows,]/sum(TCM[rows,])
        }
        TCMList[[timeCount+1]] <- TCM
        TPMList[[timeCount+1]] <- TPMEst
}

#Compare observed and expected graphically

ExpDistr <- t(as.matrix(rep(0,ncol(TPMSim))))
ExpDistr[InitialState] <- 1
TrueDistr <- ExpDistr

nStates <- ncol(TPMSim)

for (time in 0:(MaxTime-1)) {
        TimePrev <- data.frame(table(NonStatMarkovChain[NonStatMarkovChain[,3]==time,2]))
        TimePrev$Obs <- TimePrev[,2]/K
        TimePrev[,1] <- as.numeric(TimePrev[,1])
        for (stateCount in 1:ncol(TPMSim)) {
                if(is.na(match(stateCount ,TimePrev[,1]))) TimePrev <- rbind(TimePrev,c(stateCount,0,0))
        }
        TimePrev <- TimePrev[order(TimePrev[,1]),]
        TimePrev[AbState,3] <- 1-sum(TimePrev[TimePrev[,1] != AbState,3])

        TPMTrue <- TPMSim
        TPMTrue[1,1] <- exp(-( (0.2*(time+1))^2.5+(0.2*time)^2.5 ))
        Rem <- 1-TPMTrue[1,1]
        TPMTrue[1,2:(AbState-1)] <- rep(Rem*0.8/(nStates-2),nStates-2)
        TPMTrue[1,AbState] <- 1-sum(TPMTrue[1,-AbState])

        TimePrev$Exp <- t(ExpDistr)
        TimePrev$True <- t(TrueDistr)
        ExpDistr <- ExpDistr%*%TPMList[[time+1]]
        TrueDistr <- TrueDistr%*%TPMTrue


        if (time == 0) PrevMatrix <- cbind(TimePrev[,c(1,3,4,5)],rep(time,nrow(TimePrev)))
        if (time > 0) PrevMatrix <- rbind(PrevMatrix,cbind(TimePrev[,c(1,3,4,5)],rep(time,nrow(TimePrev))))
}
rm(nStates)
names(PrevMatrix) <- c("State","ObsPrev","ExpPrev","TruePrev","Time")
State1 <- PrevMatrix[PrevMatrix[,1]==1,2:5]
State2 <- PrevMatrix[PrevMatrix[,1]==2,2:5]
State3 <- PrevMatrix[PrevMatrix[,1]==3,2:5]
State4 <- PrevMatrix[PrevMatrix[,1]==4,2:5]
State5 <- PrevMatrix[PrevMatrix[,1]==5,2:5]
```

```
par(mfrow=c(3,2))

plot(State1$Time,State1$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State1")
lines(State1$Time,State1$ExpPrev,col="blue")
lines(State1$Time,State1$TruePrev,col="green")

plot(State2$Time,State2$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State2",ylim=c(0,
0.08))
#plot(State2$Time,State2$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State2")
lines(State2$Time,State2$ExpPrev,col="blue")
lines(State2$Time,State2$TruePrev,col="green")

plot(State3$Time,State3$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State3",ylim=c(0,
0.04))
#plot(State3$Time,State3$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State3")
lines(State3$Time,State3$ExpPrev,col="blue")
lines(State3$Time,State3$TruePrev,col="green")

plot(State4$Time,State4$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State4",ylim=c(0,
0.06))
#plot(State4$Time,State4$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State4")
lines(State4$Time,State4$ExpPrev,col="blue")
lines(State4$Time,State4$TruePrev,col="green")

plot(State5$Time,State5$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State5")
#plot(State5$Time,State5$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State5")
lines(State5$Time,State5$ExpPrev,col="blue")
lines(State5$Time,State5$TruePrev,col="green")

#What would happen if we treat the process as stationary?

#Calculate the transition counts and probabilities

TCM <- matrix(ncol=ncol(TPMSim),nrow=nrow(TPMSim),0)

for (rcount in 2:nrow(NonStatMarkovChain) ) {
        if ( NonStatMarkovChain[rcount,1] == NonStatMarkovChain[rcount-1,1] )  {
                TCM[NonStatMarkovChain[rcount-1,2],NonStatMarkovChain[rcount,2]] <-
TCM[NonStatMarkovChain[rcount-1,2],NonStatMarkovChain[rcount,2]] + 1
        }
}
TCM[AbState,AbState] <- 1

TPMHat <- TCM

for (i in 1:nrow(TPMSim)) {
        TPMHat[i,] <- TCM[i,]/sum(TCM[i,])
}

TPMHat

#Make observed vs Expected plot
maxTime <- max(NonStatMarkovChain[,3])
ExpDistr <- t(as.matrix(rep(0,ncol(TPMSim))))
ExpDistr[InitialState] <- 1
```

```
for (time in 0:maxTime) {
        TimePrev <- data.frame(table(NonStatMarkovChain[NonStatMarkovChain[,3]==time,2]))
        TimePrev$Obs <- TimePrev[,2]/K
        TimePrev[,1] <- as.numeric(TimePrev[,1])
        for (stateCount in 1:ncol(TPMSim)) {
                if(is.na(match(stateCount ,TimePrev[,1]))) TimePrev <- rbind(TimePrev,c(stateCount,0,0))
        }
        TimePrev <- TimePrev[order(TimePrev[,1]),]
        TimePrev[AbState,3] <- 1-sum(TimePrev[TimePrev[,1] != AbState,3])

        TimePrev$Exp <- t(ExpDistr)
        ExpDistr <- ExpDistr%*%TPMHat
        if (time == 0) PrevMatrix <- cbind(TimePrev[,c(1,3,4)],rep(time,nrow(TimePrev)))
        if (time > 0) PrevMatrix <- rbind(PrevMatrix,cbind(TimePrev[,c(1,3,4)],rep(time,nrow(TimePrev))))
}
names(PrevMatrix) <- c("State","ObsPrev","ExpPrev","Time")
State1 <- PrevMatrix[PrevMatrix[,1]==1,2:4]
State2 <- PrevMatrix[PrevMatrix[,1]==2,2:4]
State3 <- PrevMatrix[PrevMatrix[,1]==3,2:4]
State4 <- PrevMatrix[PrevMatrix[,1]==4,2:4]
State5 <- PrevMatrix[PrevMatrix[,1]==5,2:4]

par(mfrow=c(3,2))

plot(State1$Time,State1$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State1")
lines(State1$Time,State1$ExpPrev,col="blue")

plot(State2$Time,State2$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State2")
lines(State2$Time,State2$ExpPrev,col="blue")

plot(State3$Time,State3$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State3")
lines(State3$Time,State3$ExpPrev,col="blue")

plot(State4$Time,State4$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State4")
lines(State4$Time,State4$ExpPrev,col="blue")

plot(State5$Time,State5$ObsPrev,type="p",col="red",xlab="Time",ylab="Prevalence",main="State5")
lines(State5$Time,State5$ExpPrev,col="blue")

####################Kalbfleisch estimator#####################
#Example 7

#Start with 3 state process

#Q <- rbind(c(-0.25,0.1,0.15),c(0.1,-0.25,0.15),c(0.1,1.2,-1.3))
#Q <- rbind(c(-0.25,0.1,0.15),c(0.1,-0.25,0.15),c(0,0,0))
QSim <- rbind(c(-0.35,0.15,0.2),c(0.2,-0.4,0.2),c(0,0,0))
QSim <- rbind(c(-1.13,0.2,0.8,0.03,0.1),c(0.8,-1.11,0.2,0.01,0.1),c(0.4,0.9,-1.42,0.02,0.1),c(1.3,0.8,0.5,-
2.68,0.08),c(0,0,0,0,0))

NumPatients <- 1000
DeathState <- ncol(QSim)
nStates <- ncol(QSim)
```

```
InState <- 1
MaxTime <- 10

JumpData <-
RecordJumpProcess(Q=QSim,k=NumPatients,InState=InState,AbState=DeathState,T=MaxTime,saad=1010)

ObTimes <- c(0.5,1.2,2,2.5,3.1,4,5.1,5.7,6.9)

#Get state of the process at observation times

for (Pat in 1:NumPatients) {
    PatData <- JumpData[JumpData[,1]==Pat,]
        PatDiscr <- c(Pat,InState,0)
        for (time in ObTimes) {
            Pos <- PatData[PatData[,3]<=time,]
            if (length(Pos) == 3) LastState <- c(Pat,Pos[2],time)
            if (length(Pos) > 3 ) LastState <- Pos[nrow(Pos),]
            PatDiscr <- rbind(PatDiscr,c(Pat,LastState[2],time))
        }
    if (Pat == 1) DiscrObs <- PatDiscr
    if (Pat > 1) DiscrObs <- rbind(DiscrObs,PatDiscr)
}


#Calculate transition counts (Nijl) at observed time points

#MaxTime <- ObTimes[length(ObTimes)]
CountInfo <- vector("list",length(ObTimes))

for (timeCount in 1:length(ObTimes) ) {
    CurrentTime <- ObTimes[timeCount]
    PrevTime <- ifelse (timeCount ==1, 0, ObTimes[timeCount-1])

    TimeData <- DiscrObs[DiscrObs[,3]==CurrentTime | DiscrObs[,3]==PrevTime,]
    TCM <- matrix(0,ncol=ncol(QSim),nrow=nrow(QSim))
    for (rcount in 1:(nrow(TimeData)-1) ) {
        if ( TimeData[rcount,1] == TimeData[rcount+1,1] )  {
            TCM[TimeData[rcount,2],TimeData[rcount+1,2]] <-
TCM[TimeData[rcount,2],TimeData[rcount+1,2]] + 1
        }
    }
    CountInfo[[timeCount]] <- TCM
}

#Check CountInfo

NTestSum <- matrix(0,nrow=ncol(QSim),ncol=ncol(QSim))
for (i in 1:length(CountInfo) ) {
    NTestSum <- NTestSum + CountInfo[[i]]
}

library(msm)
NTestSum
ObsDF <- data.frame(DiscrObs)
names(ObsDF) <- c("PatID","State","Time")
QShape <- rbind(c(-1,0.2,0.8),c(0.8,-1,0.2),c(0,0,0))
QShape <- rbind(c(-1.7,0.2,0.8,0.3,0.4),c(0.8,-1.2,0.2,0.1,0.1),c(0.8,0.2,-1.2,0.1,0.1),c(0.8,0.1,0.2,-
1.2,0.1),c(0,0,0,0,0))

statetable.msm(State,PatID,data=ObsDF)
crudeinits.msm(State~Time,PatID,data=ObsDF,qmatrix=QShape)

ADerT <- vector("list",4)
ADerT[[1]] <- rbind(c(-1,1,0),rep(0,3),rep(0,3))
ADerT[[2]] <- rbind(c(-1,0,1),rep(0,3),rep(0,3))
ADerT[[3]] <- rbind(rep(0,3),c(1,-1,0),rep(0,3))
ADerT[[4]] <- rbind(rep(0,3),c(0,-1,1),rep(0,3))
#ADerT[[5]] <- rbind(rep(0,3),rep(0,3),c(1,0,-1))
#ADerT[[6]] <- rbind(rep(0,3),rep(0,3),c(0,1,-1))

ADerT <- vector("list",16)
ADerT[[1]] <- rbind(c(-1,1,0,0,0),rep(0,5),rep(0,5),rep(0,5),rep(0,5))
ADerT[[2]] <- rbind(c(-1,0,1,0,0),rep(0,5),rep(0,5),rep(0,5),rep(0,5))
ADerT[[3]] <- rbind(c(-1,0,0,1,0),rep(0,5),rep(0,5),rep(0,5),rep(0,5))
ADerT[[4]] <- rbind(c(-1,0,0,0,1),rep(0,5),rep(0,5),rep(0,5),rep(0,5))

ADerT[[5]] <- rbind(rep(0,5),c(-1,1,0,0,0),rep(0,5),rep(0,5),rep(0,5))
ADerT[[6]] <- rbind(rep(0,5),c(-1,0,1,0,0),rep(0,5),rep(0,5),rep(0,5))
ADerT[[7]] <- rbind(rep(0,5),c(-1,0,0,1,0),rep(0,5),rep(0,5),rep(0,5))
ADerT[[8]] <- rbind(rep(0,5),c(-1,0,0,0,1),rep(0,5),rep(0,5),rep(0,5))

ADerT[[9]] <- rbind(rep(0,5),rep(0,5),c(-1,1,0,0,0),rep(0,5),rep(0,5))
ADerT[[10]] <- rbind(rep(0,5),rep(0,5),c(-1,0,1,0,0),rep(0,5),rep(0,5))
ADerT[[11]] <- rbind(rep(0,5),rep(0,5),c(-1,0,0,1,0),rep(0,5),rep(0,5))
ADerT[[12]] <- rbind(rep(0,5),rep(0,5),c(-1,0,0,0,1),rep(0,5),rep(0,5))

ADerT[[13]] <- rbind(rep(0,5),rep(0,5),rep(0,5),c(-1,1,0,0,0),rep(0,5))
ADerT[[14]] <- rbind(rep(0,5),rep(0,5),rep(0,5),c(-1,0,1,0,0),rep(0,5))
ADerT[[15]] <- rbind(rep(0,5),rep(0,5),rep(0,5),c(-1,0,0,1,0),rep(0,5))
ADerT[[16]] <- rbind(rep(0,5),rep(0,5),rep(0,5),c(-1,0,0,0,1),rep(0,5))

GFunction <- function(B) {
        G <- vector("list",16)
        for (i in 1:16) {
                G[[i]] <- solve(B)%*%ADerT[[i]]%*%B
        }
        return(G)
}

VFunction <- function(G,D,t) {
        V <- vector("list",16)
        for (i in 1:16) {
                V[[i]] <- matrix(0,ncol=ncol(QSim),nrow=nrow(QSim))
                for (j in 1:ncol(QSim) ) {
                        for (k in 1:ncol(QSim) ) {
                                if (j != k) V[[i]][j,k] <- G[[i]][j,k]*( exp(D[j]*t) - exp(D[k]*t) )/(D[j] - D[k])

                                if (j == k) V[[i]][j,k] <- G[[i]][j,k]*t*exp(D[j]*t)
                        }
                }
        }
        return(V)
}


SFunction <- function(QEst,G,D,B) {
    S <- matrix(0,nrow=16,ncol=1)
    for (ParNum in 1:16) {
        som <- 0
        for (L in 1: length(ObTimes)) {
            for (j in 1:ncol(QEst) ) {
                for (k in 1:ncol(QEst) ) {
                    time <- ifelse(L == 1, ObTimes[L],ObTimes[L]-ObTimes[L-1])
                    Pjk <- pMatrix(QEst,nlim=30,t=time)[j,k]
                    Der <- (B%*%VFunction(G,D,time)[[ParNum]]%*%solve(B))[j,k]
                    #if (Pjk > 0.0001) som <- som + CountInfo[[L]][j,k]*Der / Pjk
                    som <- som + CountInfo[[L]][j,k]*Der / Pjk
                }
            }
        }
        S[ParNum] <- som
    }
    return(S)
}

MFunction <- function(G,D,B,QEst) {
    M <- matrix(0,nrow=16,ncol=16)
    for (u in 1:16) {
        for (v in 1:16) {
            som <- 0
            for (L in 1: length(ObTimes)) {
                for (j in 1:ncol(QEst) ) {
                    for (k in 1:ncol(QEst) ) {
                        time <- ifelse(L == 1, ObTimes[L],ObTimes[L]-ObTimes[L-
1])

                        Pjk <- pMatrix(QEst,t=time)[j,k]
                        DerU <- (B%*%VFunction(G,D,time)[[u]]%*%solve(B))[j,k]
                        DerV <- (B%*%VFunction(G,D,time)[[v]]%*%solve(B))[j,k]
                        Ni <- sum(CountInfo[[L]][j,])
                        #if (Pjk > 0.0001) som <- som +  Ni * DerU * DerV / Pjk
                        som <- som +  Ni * DerU * DerV / Pjk
                    }
                }
            }
            M[u,v] <- som
        }
    }
    return(M)
}

#ThetaCurrent <- exp(as.vector(c(0.15,0.2,0.15,0.2,0.2,1)))
ThetaCurrent <- (as.vector(c(0.20,0.15,0.15,0.25)))
ThetaCurrent <- (as.vector(rep(1,4)))
ThetaCurrent <- (as.vector(c(0.1125,0.1864,0.1452,0.1866)))
ThetaCurrent <-
(as.vector(c(0.2069,0.3508,0.0136,0.1010,0.3588,0.1711,0.0053,0.1082,0.2868,0.3869,0.0103,0.0882,0.4219,0
.1266,0.2532,0.1477)))
```

```
for (iter in 1:4) {

        #QEst <- rbind(c(-ThetaCurrent[1]-
ThetaCurrent[2],ThetaCurrent[1],ThetaCurrent[2]),c(ThetaCurrent[3],-ThetaCurrent[3]-
ThetaCurrent[4],ThetaCurrent[4]),c(ThetaCurrent[5],ThetaCurrent[6],-ThetaCurrent[5]-ThetaCurrent[6]))
        #QEst <- rbind(c(-ThetaCurrent[1]-
ThetaCurrent[2],ThetaCurrent[1],ThetaCurrent[2]),c(ThetaCurrent[3],-ThetaCurrent[3]-
ThetaCurrent[4],ThetaCurrent[4]),c(0,0,0))
        QEst <- rbind(c(-ThetaCurrent[1]-ThetaCurrent[2]-ThetaCurrent[3]-
ThetaCurrent[4],ThetaCurrent[1],ThetaCurrent[2],ThetaCurrent[3],ThetaCurrent[4]),c(ThetaCurrent[5],-
ThetaCurrent[5]-ThetaCurrent[6]-ThetaCurrent[7]-
ThetaCurrent[8],ThetaCurrent[6],ThetaCurrent[7],ThetaCurrent[8]),c(ThetaCurrent[9],ThetaCurrent[10],-
ThetaCurrent[9]-ThetaCurrent[10]-ThetaCurrent[11]-
ThetaCurrent[12],ThetaCurrent[11],ThetaCurrent[12]),c(ThetaCurrent[13],ThetaCurrent[14],ThetaCurrent[15],-
ThetaCurrent[13]-ThetaCurrent[14]-ThetaCurrent[15]-ThetaCurrent[16],ThetaCurrent[16]),c(0,0,0,0,0))

        B <- eigen(QEst,symmetric=FALSE)$vectors
        D <- eigen(QEst,symmetric=FALSE)$values
        G <- GFunction(B)
        ThetaNew <- ThetaCurrent + solve(MFunction(G,D,B,QEst))%*%SFunction(QEst,G,D,B)
        print(round(ThetaNew,4))
        #if (is.complex(ThetaNew)) ThetaNew <- Re(ThetaNew)

        ThetaCurrent <- ThetaNew
}
ThetaCurrent

ThetaCurrent <- as.vector(c(0.21693,0.09228167,0,0.36539406))

#Try example in article

rm(list=ls(all=TRUE))

NSample <- vector("list",4)

NSample[[1]] <- rbind(c(93,3,2),c(0,8,10),c(0,1,8))
NSample[[2]] <- rbind(c(89,2,2),c(0,7,5),c(0,5,15))
NSample[[3]] <- rbind(c(83,3,3),c(0,9,5),c(0,2,20))
NSample[[4]] <- rbind(c(76,3,4),c(0,6,8),c(0,0,28))

ADerT <- vector("list",3)
ADerT[[1]] <- rbind(c(-1,1,0),rep(0,3),rep(0,3))
ADerT[[2]] <- rbind(rep(0,3),c(0,-1,1),rep(0,3))
ADerT[[3]] <- rbind(rep(0,3),rep(0,3),c(0,1,-1))

Times <- c(0.15,0.6,0.35,0.8)

GFunction <- function(B) {
        G <- vector("list",3)
        for (i in 1:3) {
                G[[i]] <- solve(B)%*%ADerT[[i]]%*%B
        }
        return(G)
}
```

```
VFunction <- function(G,D,t) {
        nStates <- 3
        V <- vector("list",3)
        for (i in 1:3) {
                V[[i]] <- matrix(0,ncol=nStates,nrow=nStates)
                for (j in 1:nStates) {
                        for (k in 1:nStates) {
                                if (j != k) V[[i]][j,k] <- G[[i]][j,k]*( exp(D[j]*t) - exp(D[k]*t) )/(D[j] - D[k])

                                if (j == k) V[[i]][j,k] <- G[[i]][j,k]*t*exp(D[j]*t)
                        }
                }
        }
        return(V)
}

SFunction <- function(QEst,G,D,B) {

        S <- matrix(0,nrow=3,ncol=1)
        for (ParNum in 1:3) {
                som <- 0
                for (L in 1:length(Times)) {
                        time <- Times[L]
                        for (j in 1:ncol(QEst) ) {
                                for (k in 1:ncol(QEst) ) {
                                        Pjk <- pMatrix(QEst,nlim=30,t=time)[j,k]
                                        Der <- (B%*%VFunction(G,D,time)[[ParNum]]%*%solve(B))[j,k]
                                        if (Pjk > 0.0001) som <- som + NSample[[L]][j,k]*Der / Pjk
                                }
                        }
                }
                S[ParNum] <- som
        }
        return(S)
}

MFunction <- function(G,D,B,QEst) {
        M <- matrix(0,nrow=3,ncol=3)
        for (u in 1:3) {
                for (v in 1:3) {
                        som <- 0
                        for (L in 1:length(Times)) {
                                for (j in 1:ncol(QEst) ) {
                                        for (k in 1:ncol(QEst) ) {
                                                time <- Times[L]
                                                Pjk <- pMatrix(QEst,t=time)[j,k]
                                                DerU <- (B%*%VFunction(G,D,time)[[u]]%*%solve(B))[j,k]
                                                DerV <- (B%*%VFunction(G,D,time)[[v]]%*%solve(B))[j,k]
                                                Ni <- sum(NSample[[L]][j,])
                                                if (Pjk > 0.0001) som <- som +  Ni * DerU * DerV / Pjk
                                        }
                                }
                        }
                        M[u,v] <- som
```

```
                    }
            }
            return(M)
}

ThetaCurrent <- as.vector(c(0.2,1.26,0.27))

for (iter in 1:15) {

            QEst <- rbind(c(-ThetaCurrent[1],ThetaCurrent[1],0),c(0,-
ThetaCurrent[2],ThetaCurrent[2]),c(0,ThetaCurrent[3],-ThetaCurrent[3]))
            B <- eigen(QEst,symmetric=FALSE)$vectors
            D <- eigen(QEst,symmetric=FALSE)$values
            G <- GFunction(B)
            ThetaNew <- ThetaCurrent + solve(MFunction(G,D,B,QEst))%*%SFunction(QEst,G,D,B)

            #if (is.complex(ThetaNew)) ThetaNew <- Re(ThetaNew)
            #ThetaNew[ThetaNew < 1] <- 1
            print(ThetaNew)
            ThetaCurrent <- ThetaNew
}
exp(ThetaCurrent)



QHat <- rbind(c(-0.136,0.136,0),c(0,-2.28,2.28),c(0,0.470,-0.470))
pMatrix(QHat,nlim=30,t=0.15)[3,3]*9


#####################Kay estimator######################
#Examples 8 - 10

#Start with 3 state process and with no censoring to see if function works

QSim <- rbind(c(-0.35,0.15,0.2),c(0.2,-0.4,0.2),c(0,0,0))
QSim <- rbind(c(-1.13,0.2,0.8,0.03,0.1),c(0.8,-1.11,0.2,0.01,0.1),c(0.4,0.9,-1.42,0.02,0.1),c(1.3,0.8,0.5,-
2.68,0.08),c(0,0,0,0,0))

NumPatients <- 1000
DeathState <- ncol(QSim)
nStates <- ncol(QSim)
InState=1
NumPatients <- 1000
MaxTime <- 50

JumpData <-
RecordJumpProcess(Q=QSim,AbState=DeathState,InState=InState,k=NumPatients,T=MaxTime,saad=1010)

#Extract discrete time point observations with death being observed within one day

for (Pat in 1:NumPatients) {
            PatData <- JumpData[JumpData[,1]==Pat,]
            PatObs <- c(Pat,InState,0)
            CumTime <- 0
            sent  <- 1
            while (sent ==1 ) {
                    NewTime <- runif(1,min=0.5,max=1.5)
                    #NewTime <- max(rexp(1,rate=1),1/365)
                    CumTime <- CumTime + NewTime
                    Pos <- PatData[PatData[,3]<=CumTime,]
                    if (length(Pos) == 3) LastState <- Pos
                    if (length(Pos) > 3 ) LastState <- Pos[nrow(Pos),]
                    if (LastState[2] != DeathState) LastState[3] <- CumTime
                    PatObs <- rbind(PatObs,LastState)
                    if (LastState[2] == DeathState | CumTime > MaxTime) sent <- 0
            }
            if (Pat == 1) DiscrObs <- PatObs
            if (Pat > 1) DiscrObs <- rbind(DiscrObs,PatObs)
}

#Extract discrete time point observations with death not necesarily being observed within one day

for (Pat in 1:NumPatients) {
            PatData <- JumpData[JumpData[,1]==Pat,]
            PatObs <- c(Pat,InState,0)
            CumTime <- 0
            sent  <- 1
            while (sent ==1 ) {
                    NewTime <- runif(1,min=0.5,max=1.5)
                    #NewTime <- rexp(1,rate=1)
                    CumTime <- CumTime + NewTime
                    Pos <- PatData[PatData[,3]<=CumTime,]
                    if (length(Pos) == 3) LastState <- Pos
                    if (length(Pos) > 3 ) LastState <- Pos[nrow(Pos),]
                    LastState[3] <- CumTime
                    PatObs <- rbind(PatObs,LastState)
                    if (LastState[2] == DeathState | CumTime > MaxTime) sent <- 0
            }
            if (Pat == 1) DiscrObsNoDeath <- PatObs
            if (Pat > 1) DiscrObsNoDeath <- rbind(DiscrObsNoDeath,PatObs)
}

likeKayNoCenMatrixNLM <- function(As) {

            QEst <- matrix(0,nrow=5,ncol=5)
            QEst[1,2] <- As[1]
            QEst[1,3] <- As[2]
            QEst[1,1] <- -sum(QEst[1,-1])

            QEst[2,1] <- As[3]
            QEst[2,3] <- As[4]
            QEst[2,2] <- -sum(QEst[2,-2])

            like <- 0
            for (Pat in 1:NumPatients) {
                    PatData <- DiscrObsNoDeath[DiscrObsNoDeath[,1] == Pat,]
                    for (rcount in 2:nrow(PatData) ) {
                            tyd <- PatData[rcount,3] - PatData[rcount-1,3]
                            TPMtemp <- pMatrix(QEst,nlim=30,t=tyd)
                            Add <- TPMtemp[PatData[rcount-1,2],PatData[rcount,2]]
                            like <- like + log(Add)
                    }
```

```
		}
		return(-like)
}

likeKayNoCenPFunctionNLM <- function(As) {

		Alpha <- matrix(0,ncol=3,nrow=3)
		Alpha[1,2] <- As[1]
		Alpha[1,3] <- As[2]
		Alpha[1,1] <- -sum(Alpha[1,-1])

		Alpha[2,1] <- As[3]
		Alpha[2,3] <- As[4]
		Alpha[2,2] <- -sum(Alpha[2,-2])

		R <- c(-Alpha[1,1],-Alpha[2,2])

		Lambda1 <- -(R[1]+R[2] + sqrt( (R[1]-R[2])^2 + 4*As[1]*As[2] ) )/2
		Lambda2 <- -(R[1]+R[2] - sqrt( (R[1]-R[2])^2 + 4*As[3]*As[4] ) )/2


		like <- 0
		for (Pat in 1:NumPatients) {
				PatData <- DiscrObsNoDeath[DiscrObsNoDeath[,1] == Pat,]
				for (rcount in 2:nrow(PatData) ) {
						tyd <- PatData[rcount,3] - PatData[rcount-1,3]
						CurSt <- PatData[rcount-1,2]
						NewSt <- PatData[rcount,2]

						if (NewSt == CurSt) Add <- ( (R[-CurSt]+Lambda1)*exp(Lambda1*tyd) - (R[-
CurSt]+Lambda2)*exp(Lambda2*tyd) )/(Lambda1 - Lambda2)
						if (NewSt == DeathState) Add <-	1 + ( (Alpha[CurSt,3] +
Lambda2)*exp(Lambda1*tyd) - (Alpha[CurSt,3] + Lambda1)*exp(Lambda2*tyd) ) / (Lambda1 - Lambda2)
						if (NewSt != DeathState & NewSt != CurSt) Add <- (
Alpha[CurSt,NewSt]*(exp(Lambda1*tyd) - exp(Lambda2*tyd)) ) / (Lambda1-Lambda2)

						like <- like + log(Add)
				}

		}
		return(-like)
}

likeKayWithCenMatrixNLM <- function(As) {

		QEst <- matrix(0,nrow=5,ncol=5)
		QEst[1,2] <- As[1]
		QEst[1,3] <- As[2]
		QEst[1,1] <- -sum(QEst[1,-1])

		QEst[2,1] <- As[3]
		QEst[2,3] <- As[4]
		QEst[2,2] <- -sum(QEst[2,-2])
```

```
    PosStates <- 1:ncol(QEst)
    like <- 0
    for (Pat in 1:NumPatients) {
        PatData <- DiscrObs[DiscrObs[,1] == Pat,]
        for (rcount in 2:nrow(PatData) ) {
            if (PatData[rcount,2] != DeathState) {
                tyd <- PatData[rcount,3] - PatData[rcount-1,3]
                TPMtemp <- pMatrix(QEst,nlim=30,t=tyd)
                Add <- TPMtemp[PatData[rcount-1,2],PatData[rcount,2]]
                like <- like + log(Add)
            }
            if (PatData[rcount,2] == DeathState) {
                tyd <- PatData[rcount,3] - PatData[rcount-1,3]
                TPMMissingJump <- pMatrix(QEst,nlim=30,t=tyd-1/365)
                TPMDeathJump <- pMatrix(QEst,nlim=30,t=1/365)
                for (stateCount in PosStates[PosStates!=DeathState] ) {
                    Add <- TPMMissingJump[PatData[rcount-
1,2],stateCount]*TPMDeathJump[stateCount,DeathState]
                    like <- like + log(Add)
                }
            }
        }

    }
    return(-like)

}

P13 <- function(A13,lambda1,lambda2,tyd) {
    PR <- 1 + ( (A13+lambda2)*exp(lambda1*tyd) - (A13+lambda1)*exp(lambda2*tyd) ) / (lambda1-
lambda2)
    return(PR)
}

P23 <- function(A23,lambda1,lambda2,tyd) {
    PR <- 1 + ( (A23+lambda2)*exp(lambda1*tyd) - (A23+lambda1)*exp(lambda2*tyd) ) / (lambda1-
lambda2)
    return(PR)
}

P11 <- function(R2,lambda1,lambda2,tyd) {
    PR <- ( (R2+lambda1)*exp(lambda1*tyd) - (R2+lambda2)*exp(lambda2*tyd) )/(lambda1 - lambda2)

    return(PR)
}

P22 <- function(R1,lambda1,lambda2,tyd) {
    PR <- ( (R1+lambda1)*exp(lambda1*tyd) - (R1+lambda2)*exp(lambda2*tyd) )/(lambda1 - lambda2)
    return(PR)
}



P12 <- function(A12,lambda1,lambda2,tyd) {
    PR <- (A12*(exp(lambda1*tyd) - exp(lambda2*tyd)) ) / (lambda1-lambda2)
```

```
        return(PR)
}

P21 <- function(A21,lambda1,lambda2,tyd) {
        PR <- (A21*(exp(lambda1*tyd) - exp(lambda2*tyd)) ) / (lambda1-lambda2)
        return(PR)
}

likeKayWithCenPFunctionNLM <- function(As) {

        A12 <- As[1]
        A13 <- As[2]
        A21 <- As[3]
        A23 <- As[4]

        R <- c(A12+A13,A21+A23)

        Lambda1 <- -(R[1]+R[2] + sqrt( (R[1]-R[2])^2 + 4*A12*A21 ) )/2
        Lambda2 <- -(R[1]+R[2] - sqrt( (R[1]-R[2])^2 + 4*A12*A21 ) )/2


        like <- 0
        for (Pat in 1:NumPatients) {
                PatData <- DiscrObs[DiscrObs[,1] == Pat,]
                for (rcount in 2:nrow(PatData) ) {
                        tyd <- PatData[rcount,3] - PatData[rcount-1,3]
                        CurSt <- PatData[rcount-1,2]
                        NewSt <- PatData[rcount,2]

                        if (NewSt == CurSt) Add <- ( (R[-CurSt]+Lambda1)*exp(Lambda1*tyd) - (R[-
CurSt]+Lambda2)*exp(Lambda2*tyd) )/(Lambda1 - Lambda2)

                        if (CurSt == 1 & NewSt == 3) Add <- P11(R[1],Lambda1,Lambda2,tyd-
1/365)*P13(A13,Lambda1,Lambda2,1/365)+P12(A12,Lambda1,Lambda2,tyd-
1/365)*P23(A23,Lambda1,Lambda2,1/365)
                        if (CurSt == 2 & NewSt == 3) Add <- P21(A21,Lambda1,Lambda2,tyd-
1/365)*P13(A13,Lambda1,Lambda2,1/365)+P22(R[2],Lambda1,Lambda2,tyd-
1/365)*P23(A23,Lambda1,Lambda2,1/365)

                        if (CurSt == 1 & NewSt == 2) Add <- P12(A12,Lambda1,Lambda2,tyd)
                        if (CurSt == 2 & NewSt == 1) Add <- P21(A21,Lambda1,Lambda2,tyd)

                        #if (is.na(log(Add))) print (Pat)
                        like <- like + log(Add)
                        rm(Add)
                }

        }
        return(-like)
}


likeJacWithCenMatrixNLM <- function(As) {
        QEst <- matrix(0,nrow=5,ncol=5)
        QEst[1,2] <- As[1]
        QEst[1,3] <- As[2]
```

```
    QEst[1,1] <- -sum(QEst[1,-1])

    QEst[2,1] <- As[3]
    QEst[2,3] <- As[4]
    QEst[2,2] <- -sum(QEst[2,-2])

    PosStates <- 1:ncol(QEst)

    like <- 0
    for (Pat in 1:NumPatients) {
        PatData <- JumpData[JumpData[,1] == Pat,]
        for (rcount in 2:nrow(PatData) ) {
            if (PatData[rcount,2] != DeathState) {
                tyd <- PatData[rcount,3] - PatData[rcount-1,3]
                TPMtemp <- pMatrix(QEst,nlim=30,t=tyd)
                like <- like + log(TPMtemp[PatData[rcount-1,2],PatData[rcount,2]])
            }
            if (PatData[rcount,2] == DeathState) {
                tyd <- PatData[rcount,3] - PatData[rcount-1,3]
                TPMMissingJump <- pMatrix(QEst,nlim=30,t=tyd)
                for (stateCount in PosStates[PosStates!=DeathState] ) {
                    like <- like + log(TPMMissingJump[PatData[rcount-
1,2],stateCount]*QEst[stateCount,DeathState])
                }
            }
        }

    }
    return(-like)
}

likeJacWithCenPFunctionNLM <- function(As) {

    A12 <- As[1]
    A13 <- As[2]
    A21 <- As[3]
    A23 <- As[4]

    R <- c(A12+A13,A21+A23)
    Lambda1 <- -(R[1]+R[2] + sqrt( (R[1]-R[2])^2 + 4*A12*A21 ) )/2
    Lambda2 <- -(R[1]+R[2] - sqrt( (R[1]-R[2])^2 + 4*A12*A21 ) )/2

    like <- 0
    for (Pat in 1:NumPatients) {
        PatData <- DiscrObs[DiscrObs[,1] == Pat,]
        for (rcount in 2:nrow(PatData) ) {
            tyd <- PatData[rcount,3] - PatData[rcount-1,3]
            CurSt <- PatData[rcount-1,2]
            NewSt <- PatData[rcount,2]

            if (CurSt == DeathState) print("Error")

            if (NewSt == CurSt) Add <- ( (R[-CurSt]+Lambda1)*exp(Lambda1*tyd) - (R[-
CurSt]+Lambda2)*exp(Lambda2*tyd) )/(Lambda1 - Lambda2)
```

```
                              if (CurSt == 1 & NewSt == 3) Add <-
P11(R[1],Lambda1,Lambda2,tyd)*A13+P12(A12,Lambda1,Lambda2,tyd)*A23
                              if (CurSt == 2 & NewSt == 3) Add <-
P21(A21,Lambda1,Lambda2,tyd)*A13+P22(R[2],Lambda1,Lambda2,tyd)*A23

                              if (CurSt == 1 & NewSt == 2) Add <- P12(A12,Lambda1,Lambda2,tyd)
                              if (CurSt == 2 & NewSt == 1) Add <- P21(A21,Lambda1,Lambda2,tyd)

                              like <- like + log(Add)
                              rm(Add)
                    }

          }
          return(-like)
}

#Get initial values from msm

library(msm)
ObsDF <- data.frame(DiscrObs)
ObsDFNoDeath <- data.frame(DiscrObsNoDeath)

names(ObsDF) <- c("PatID","State","Time")
names(ObsDFNoDeath) <- c("PatID","State","Time")

QShape <- rbind(c(-1,0.2,0.8),c(0.8,-1,0.2),c(0,0,0))
QShape <- rbind(c(-1.7,0.2,0.8,0.3,0.4),c(0.8,-1.2,0.2,0.1,0.1),c(0.8,0.2,-1.2,0.1,0.1),c(0.8,0.1,0.2,-
1.2,0.1),c(0,0,0,0,0))

QIn <- crudeinits.msm(State~Time,PatID,data=ObsDF,qmatrix=QShape)
QInNoDeath <- crudeinits.msm(State~Time,PatID,data=ObsDFNoDeath,qmatrix=QShape)

ptm <- proc.time()
QMSM <- msm(State~Time,PatID,data=ObsDF,qmatrix=QIn,death=3)
QMSMTime <- proc.time() - ptm

ptm <- proc.time()
QMSMNoDeath <- msm(State~Time,PatID,data=ObsDFNoDeath,qmatrix=QIn)
QMSMNoDeathTime <- proc.time() - ptm

As <- c(QIn[1,2],QIn[1,3],QIn[2,1],QIn[2,3])
AsNoDeath <- c(QInNoDeath[1,2],QInNoDeath[1,3],QInNoDeath[2,1],QInNoDeath[2,3])

As <-
c(QIn[1,2],QIn[1,3],QIn[1,4],QIn[1,5],QIn[2,1],QIn[2,3],QIn[2,4],QIn[2,5],QIn[3,1],QIn[3,2],QIn[3,4],QIn[3,5],QIn
[4,1],QIn[4,2],QIn[4,3],QIn[4,5])

library(stats4)

ptm <- proc.time()
Est1 <- nlm(likeKayNoCenMatrixNLM,p=AsNoDeath,stepmax=0.1)
Est1Time <- proc.time() - ptm

ptm <- proc.time()
Est3 <- nlm(likeKayNoCenPFunctionNLM,p=AsNoDeath,stepmax=0.1)
Est3Time <- proc.time() - ptm
```

```
ptm <- proc.time()
Est5 <- nlm(likeKayWithCenMatrixNLM,p=As,stepmax=0.1)
Est5Time <- proc.time() - ptm

ptm <- proc.time()
Est7 <- nlm(likeKayWithCenPFunctionNLM,p=As,stepmax=0.1)
Est7Time <- proc.time() - ptm

ptm <- proc.time()
Est9 <- nlm(likeJacWithCenMatrixNLM,p=As,stepmax=0.1)
Est9Time <- proc.time() - ptm

ptm <- proc.time()
Est11 <- nlm(likeJacWithCenPFunctionNLM,p=As,stepmax=0.1)
Est11Time <- proc.time() - ptm

GetMatrix <- function(vec) {
        TIM <- matrix(0,nrow=3,ncol=3)
        TIM[1,2] <- vec[1]
        TIM[1,3] <- vec[2]
        TIM[1,1] <- -sum(TIM[1,-1])

        TIM[2,1] <- vec[3]
        TIM[2,3] <- vec[4]
        TIM[2,2] <- -sum(TIM[2,-2])
        return (TIM)
}

GetMatrix <- function(vec) {
        TIM <- matrix(0,nrow=5,ncol=5)
        TIM[1,2] <- vec[1]
        TIM[1,3] <- vec[2]
        TIM[1,4] <- vec[3]
        TIM[1,5] <- vec[4]
        TIM[1,1] <- -sum(TIM[1,-1])

        TIM[2,1] <- vec[5]
        TIM[2,3] <- vec[6]
        TIM[2,4] <- vec[7]
        TIM[2,5] <- vec[8]
        TIM[2,2] <- -sum(TIM[2,-2])

        TIM[3,1] <- vec[9]
        TIM[3,2] <- vec[10]
        TIM[3,4] <- vec[11]
        TIM[3,5] <- vec[12]
        TIM[3,3] <- -sum(TIM[3,-3])

        TIM[4,1] <- vec[13]
        TIM[4,2] <- vec[14]
        TIM[4,3] <- vec[15]
        TIM[4,5] <- vec[16]
        TIM[4,4] <- -sum(TIM[4,-4])
        return (TIM)
}
```

```
GetMatrix(As)
GetMatrix(Est1$estimate); Est1Time
GetMatrix(Est3$estimate); Est3Time
GetMatrix(Est5$estimate); Est5Time
GetMatrix(Est7$estimate); Est7Time
GetMatrix(Est9$estimate); Est9Time
GetMatrix(Est11$estimate); Est11Time

SSD1 <- GetMatrix(Est1$estimate) - QSim; diag(SSD1) <- 0; sum(SSD1^2)
SSD3 <- GetMatrix(Est3$estimate) - QSim; diag(SSD3) <- 0; sum(SSD3^2)
SSD5 <- GetMatrix(Est5$estimate) - QSim; diag(SSD5) <- 0; sum(SSD5^2)
SSD7 <- GetMatrix(Est7$estimate) - QSim; diag(SSD7) <- 0; sum(SSD7^2)
SSD9 <- GetMatrix(Est9$estimate) - QSim; diag(SSD9) <- 0; sum(SSD9^2)
SSD11 <- GetMatrix(Est11$estimate) - QSim; diag(SSD11) <- 0; sum(SSD11^2)

sum((GetMatrix(Est1$estimate) - QSim)^2)
sum((GetMatrix(Est3$estimate) - QSim)^2)
sum((GetMatrix(Est5$estimate) - QSim)^2)
sum((GetMatrix(Est7$estimate) - QSim)^2)
sum((GetMatrix(Est9$estimate) - QSim)^2)
sum((GetMatrix(Est11$estimate) - QSim)^2)

QMSMNoDeath

SSDQMSM <- QMSM$Qmatrices$baseline - QSim; diag(SSDQMSM) <- 0; sum(SSDQMSM^2)
SSDQMSMNoDeath <- QMSMNoDeath$Qmatrices$baseline - QSim; diag(SSDQMSMNoDeath) <- 0;
sum(SSDQMSMNoDeath^2)

ptm <- proc.time()
Est1_5state <- nlm(likeKayNoCenMatrixNLM,p=As,stepmax=0.1)
Est1Time_5state <- proc.time() - ptm

ptm <- proc.time()
Est5_5state <- nlm(likeKayWithCenMatrixNLM,p=As,stepmax=0.1)
Est5Time_5state <- proc.time() - ptm

ptm <- proc.time()
Est9_5state <- nlm(likeJacWithCenMatrixNLM,p=As,stepmax=0.1)
Est9Time_5state <- proc.time() - ptm

QMSM_5state <- QMSM
QMSMNoDeath_5State <- QMSMNoDeath



#######Example 11: 5 state process with age and gender effect with transitions not observed exactly

QSim <- rbind(c(-1.4,0.2,0.8,0.3,0.1),c(0.8,-1.2,0.2,0.1,0.1),c(0.4,0.9,-1.6,0.2,0.1),c(0.2,0.4,0.3,-
1.7,0.8),c(0,0,0,0,0))

NumPatients <- 1000
MaxTime <- 50

#Generate gender
```

```
set.seed(1604)
Gender <- rbinom(n=NumPatients,size=1,prob=0.7)
GenderEffect <- 0.4

#Generate age
set.seed(1604)
Age <- runif(n=NumPatients,min=0,max=100)
AgeEffect <- 0.05

DeathState <- ncol(QSim)
nStates <- ncol(QSim)
InState <- 1

JumpData <-
RecordJumpProcessGenderAgeEffect(Q=QSim,GenderVector=Gender,GenderEffect=GenderEffect,AgeVector=
Age,AgeEffect=AgeEffect,AbState=DeathState,InState=1,k=NumPatients,T=MaxTime,saad=1010)


#Extract discrete time point observations with death being observed within one day

for (Pat in 1:NumPatients) {
        PatData <- JumpData[JumpData[,1]==Pat,]
        PatObs <- c(Pat,InState,0)
        CumTime <- 0
        sent  <- 1
        while (sent ==1 ) {
                NewTime <- runif(1,min=0.5,max=1.5)
                CumTime <- CumTime + NewTime
                Pos <- PatData[PatData[,3]<=CumTime,]
                if (length(Pos) == 3) LastState <- Pos
                if (length(Pos) > 3 ) LastState <- Pos[nrow(Pos),]
                if (LastState[2] != DeathState) LastState[3] <- CumTime
                PatObs <- rbind(PatObs,LastState)
                if (LastState[2] == DeathState | CumTime > MaxTime) sent <- 0
        }
        if (Pat == 1) DiscrObs <- PatObs
        if (Pat > 1) DiscrObs <- rbind(DiscrObs,PatObs)
}

library(msm)

ObsDF <- data.frame(DiscrObs)
names(ObsDF) <- c("PatID","State","Time")
ObsDF$Sex <- Gender[ObsDF[,1]]
ObsDF$Age <- Age[ObsDF[,1]]

FirstQ <- rbind(c(-1.5,0.2,0.8,0.3,0.5),c(0.8,-1.4,0.2,0.3,0.1),c(0.8,0.1,-1.3,0.3,0.1),c(0.8,0.2,0.3,-
1.4,0.1),c(0,0,0,0,0))

QmsmIn <- crudeinits.msm(State~Time,PatID,data=ObsDF,qmatrix=FirstQ )

ptm <- proc.time()
Qmsm <- msm(State~Time,PatID,data=ObsDF,qmatrix=QmsmIn,death=5,covariates=~Sex+Age)
proc.time() - ptm

qmatrix.msm(Qmsm,covariates=list(Sex=0,Age=0))
```

```
Qmsm

par(mfrow=c(3,2))
plot.prevalence.msm(Qmsm, maxtime = 30)

ptm <- proc.time()
LoS <- totlos.msm(Qmsm)
proc.time() - ptm

LoS[1]+LoS[2]+LoS[3]+LoS[4]

test <- DiscrObs[DiscrObs[,2]==DeathState,]
summary(test[,3])

test <- JumpData[JumpData[,2]==DeathState,]
summary(test[,3])

#Do pearson's test

ptm <- proc.time()
GoodFit <- pearson.msm(Qmsm,boot=TRUE)
proc.time() - ptm; #21003 seconds (5.8 hours)

ptm <- proc.time()
GoodFit2 <- pearson.msm(Qmsm,boot=TRUE,covgroups=1)
proc.time() - ptm
```

References

1. Aalen OO, Johansen S. An empirical transition matrix for non-homogenous Markov chains based on censored observations. The Scandinavian Journal of Statistics, **5**, 141 - 150 (1978).

2. Adabag AS et al. Sudden Death after Myocardial Infarction. Journal of American Medical Association, **300**:17, 2022 - 2029 (2008).

3. Aguirre-Hernandez R and Farewell VT. A Pearson-type goodness-of-fit test for stationary and time-continuous Markov regression models. Statistics in Medicine, **21**, 1899 - 1911 (2002).

4. Albert A. Estimating the infinitesimal generator of a continuous time, finite state Markov Process. Annals of Mathematical Statistics, **33,** 727 - 753 (1962).

5. Andersen PK, Hansen LS, Keiding N. Non- and semi-parametric estimation of transition probabilities from censored observation of a non-homogeneous Markov process. The Scandinavian Journal of Statistics, **18**:2, 153 - 167 (1991).

6. Andersen PK. Multistate models in survival analysis: a study of nephropathy and mortality in diabetes. Statistics in Medicine, **7**:6, 661 - 670 (1988).

7. Anderson M. Option pricing using hidden Markov models. Thesis (M.Sc. (Mathematics of Finance)), University of Cape Town (2006).

8. Anderson TW, Goodman LA. Statistical inference about Markov chains. The Annals of Mathematical Statistics, **28**:1, 89 - 110 (1957).

9. Argesti A. Categorical Data Analysis, Second Edition. Wiley Interscience, New Jersey (2002).

10. Billard L, Meshkani MR. Estimation of a stationary Markov Chain. Journal of the American Statistical Association, **90**:429, 307 - 315 (1995).

11. Brummer JNL. A phoneme-based hidden Markov model for speaker independent acoustic-phonetic alignment. Thesis (M.Ing.), University of Stellenbosch (1998).

12. Bureau A, Shiboski S and Hughes JP. Applications of continuous time hidden Markov models to the study of misclassified disease outcomes. Statistics in Medicine, **22**, 441 - 462 (2003).

13. Chakraborti, S, Eryilmaz, S, Human SW. A phase II nonparametric control chart based on precedence statistics with runs-type signalling rules. Computational Statistics and Data Analysis, **53**:4, 1054 - 1065 (2009).

14. Chang CM et al. Estimation and prediction system for multi-state disease process: application to analysis of organized screening regime. Journal of Evaluation in Clinical Practice, **13**, 867 - 881 (2007).

15. Chen HH, Duffy SW, Tabar L. A Markov chain method to estimate the tumour progression rate from preclinical phase, sensitivity and positive predictive value for mammography in breast cancer screening. The Statistician, **45**:3, 307 - 317 (1996).

16. Cinlar E. Introduction to Stochastic Processes. Prentice-Hall, Inc., Englewood Cliffs, N.J (1975).

17. Coetzer J. Off-line signature verification. Thesis (PhD), University of Stellenbosch (2005).

18. Collet D. Modelling survival data in medical research, Texts in Statistical Science.. Chapman & Hall. (1994).

19. Cox DR. Regression Models and Life-Tables. Journal of the Royal Statistical Society. Series B (Methodological), **34**:2, 187 - 220 (1972).

20. Cox DR, Miller HD. The Theory of Stochastic Processes. Methuen & Co Ltd (1965).

21. Craig BA et al. A Bayesian approach to modelling the natural history of a chronic condition from observations with interventions. Statistics in Medicine, **18**, 1355 - 1371 (1999).

22. Craig BA, Sendi PP. Estimation of the transition matrix of a discrete-time Markov chain. Health Economics, **11**, 33 - 42, (2002).

23. D'Agostino, Wolf, Belanger. http://www.framinghamheartstudy.org/risk/stroke.html. Accessed 16 March 2010

24. Danaher PJ. A Markov Chain Model for Magazine Exposure; Journal of the American Statistics Association **84**, 922 - 926 (1989).

25. David JC. Statistics and Data Analysis in Geology. New York: John Wiley (1986).

26. Doob JL, Stochastic Processes. Wiley New York (1953).

27. Du Toit E. Modelling the co-infection dynamics of HIV-1 and M. Tuberculosis; Thesis (M.Eng (Electronic engineering)), University of Pretoria (2008).

28. Du Preez JA. Efficient higher-order hidden Markov modelling. Dissertation (PhD.), University of Stellenbosch (1997).

29. Duffy SW, Chen HH. Estimation of mean sojourn time in breast cancer screening using a Markov chain model of both entry to and exit from the preclinical detectable phase. Statistics in Medicine, **14**, 1531 - 1543 (1995).

30. Ehrenberg ASC. An appraisal of Markov Brand-Switching Models. Journal of Marketing Research, **11**, 347 - 262 (1965).

31. El-Asfouri S, McInnis BC, Kapadia AS. Stochastic compartmental modeling and Parameter estimation with application to Cancer treatment follow-up studies. Bulletin of Mathematical Biology, **41**, 203 - 215 (1979).

32. Engelbrecht HA. Efficient decoding of high-order hidden Markov models. Dissertation (PhD), University of Stellenbosch (2007).

33. Feller W. An Introduction to Probability Theory and Its Applications, Vol 1, 3rd ed. New York: Wiley (1968).

34. Ferreira MAR, Suchard MA. Bayesian Analysis of Elapsed Times in Continuous-Time Markov Chains. Canadian Journal of Statistics, **36**:3, 355 - 368 (1979).

35. Frydman H. A nonparametric estimation procedure for a periodically observed three-state Markov process with application to AIDS. Journal of the Royal Statistical Society B, **54**, 836 - 866 (1992).

36. Gamerman D. Markov Chain Monte Carlo: Stochastic simulation for Bayesian inference. Chapman & Hall, 1st edition (1997).

37. Gautrais J, Michelena P, Sibbald A, Bon R and Deneubourg JL. Allelomimetic synchronization in Merino sheep. Animal Behaviour, **74**, 1443 - 1454 (2007).

38. Genteleman RC et al. Multi-state Markov models for analysing incomplete disease history data with illustrations for HIV disease. Statistics in Medicine, **13**:3, 805 - 821 (1994).

39. Grassly NC, Ward ME, Ferris S, Mabey DC, Bailey RL (2008). The Natural History of Trachoma Infection and Disease in a Gambian Cohort with Frequent Follow-Up. Public Library of Science, Neglected Tropical Diseases, **2**:12, e341. doi:10.1371/journal.pntd.0000341 (2008).

40. Guihenneuc-Jouyaux C, Richardson S, Longini IM. Modelling markers of disease progression by a hidden Markov process : Application to characterising CD4 cell decline. Biometrics, **56**, 733 - 741 (2000).

41. Guillermo M, Jones RH. Multi-state Markov models and diabetic retinopathy. Statistics in Medicine, **14**:18, 1975 - 1983 (1995).

42. Gumedze FN. Modelling the relationships between clinical markers of the Human Immunodeficiency Virus disease in a South African population; MBusSc (Statistical Science), University of Cape Town (1999).

43. Gupta MR. A Measure Theory Tutorial (Measure theory for Dummies). UWEE Technical report Number UWEETR-2006-0008. (2006) Available form https://www.ee.washington.edu/techsite/papers/documents/UWEETR-2006-0008.pdf. Accessed 1 March 2010.

44. Halmos PR and Savage LJ. Application of the Radon-Nikodym Theorem to the Theory of Sufficient Statistics. The Annals of Mathematical Statistics, **20**:2, p. 225 - 241 (1949).

45. Hethcote HW. The Mathematics of Infectious Diseases. Journal of the Society for Industrial and Applied Mathematics Review, **42**:4, 599 - 653 (2000).

46. Hougaard P. Multi-state models: a review. Lifetime Data Analysis, **5**, 239 - 264 (1999).

47. Hubbard RA, Inoue LYT, Fann JR. Modeling nonhomogeneous Markov processes via time transformation. Biometrics, **64**:3, 843 - 850 (2008).

48. Hui-Min W, Ming-Fang Y, Chn TH. SAS macro program for non-homogeneous Markov process in modeling multi-state disease progression. Computer Methods and Programs in Biomedicine, **75**, 95 - 105. (2004).

49. Jackson C. Multi-state modeling with R: The msm package, Version 0.9.6, 11 February 2010.

50. Jackson C. Multi-state models for panel data: the msm package for R. To be published in the Journal of Statistical Software.

51. Jackson CH et al. Multistate Markov models for disease progression with classification error. Journal of the Royal Statistical Society Series D – The Statistician, **52**:2, 193 - 209 (2003).

52. Kajama SM. Efficient decoding of high-order hidden Markov models. Thesis (M.Sc.), University of South Africa (2008).

53. Kalbfleisch JD and Lawless JF. Least-squares estimation of transition probabilities from aggregate data. The Canadian Journal of Statistics, **12**:3, 169-182 (1984).

54. Kalbfleisch JD and Lawless JF. The analysis of panel data under the Markov assumption. Journal of the American Statistical Association, **80**, 863 - 871 (1985).

55. Kalbfleisch JD, Prentice RL and Prentice Ross L. The Statistical Analysis of Failure Time Data, Wiley Series in Probability and Mathematical Statistics (1980).

56. Kaplan EL and Meier P. Nonparametric Estimation from Incomplete Observations. Journal of the American Statistical Association, **53**:282, 457- 481 (1958).

57. Kay R. A Markov model for analysing cancer markers and disease states in survival studies. Biometrics, **42**:2, 855 - 865 (1986).

58. Klotz JH, Sharpless LD. Estimation for a Markov Heart Transplant Model. Journal of the Royal Statistical Society. Series D (The Statistician), **43**:3, 431 - 438 (1994).

59. Labeodan, MMO, Stochastic analysis of AIDS epidemiology, Thesis (PhD Mathematical Statistics), University of Pretoria (2009). Accessed on the 14th of May 2010. Available from http://upetd.up.ac.za/thesis/available/etd-10172009-112824

60. Le Richie PJ. Handwritten signature verification: a hidden Markov model approach. MIng (Elektroniese Ingenieurswese), Universiteit van Stellenbosch (2001).

61. Lee TC, Judge GG, Zellner A. Estimating the Parameters of the Markov Probability Model from Aggregate Time Series Data. North Holland Publishing Company: Amsterdam, London (1970).

62. Lehmensiek R. Maximum likelihood estimation versus least-squares estimation in semi-continuous hidden Markov modelling. Thesis (M.Ing.), University of Stellenbosch (1995).

63. Lifson F. Specification and verification of systems using model checking and Markov reward models. Thesis (M.Sc. (Computer Science)), University of Cape Town (2004).

64. Limon A. Automatic recognition of dysarthric speech: a hidden Markov modelling approach. Thesis (M.Sc.(Medicine)(Biomedical Engineering)), University of Cape Town (1993).

65. Longini IM et al. Statistical analysis of the stages of HIV infection using a Markov model. Statistics in Medicine, **8**, 851 - 843 (1989).

66. Lipschultz S. Shaum's outline of theory and problems of linear algebra. Shaum's outline series. McGraw-Hill book company (1968).

67. Mabuza BR. Applied inverse scattering. Thesis (D.Phil.), University of South Africa (2005).

68. Machado LM, Suarez CC, Alvarez JU. tdc.msm: An R library for the analysis of multistate survival data. Computer Methods and Programmes in Biomedicine, **86**, 131 - 140 (2007).

69. Mariotto AB et al. Estimation of the Acquired Immunodeficiency Syndrome Incubation Period in Intravenous Drug Users: A Comparison with Male Homosexuals. The American Journal of Epidemiology, **135**, 428 - 437 (1992).

70. Marshall G, Jones RH. Multi-state Markov models and diabetic retinopathy. Statistics in Medicine, **14**, 1975 - 1983 (1995).

71. Martin JJ. Bayesian Decision Problems and Markov Chains. Publications in Operations Research, Operations research Society of America. John Wiley & Sons, Inc. (1967).

72. McGetrick MJ. Performance of the prediction control neural network and the role of temporal modelling in speaker identification. Thesis (M.Ing.), University of Stellenbosch (1994).

73. McGrath M. Markov random field image modelling. Thesis (M.Sc. (Electrical Engineering)), University of Cape Town (2003).

74. Meira-Machado L, Uña-Álvarez J, Cadarso-Suárez C and Andersen PK. Multi-state models for the analysis of time-to-event data. Statistical Methods in Medical Research, **18**, 195 - 222 (2009).

75. Mekonnen Y, Geskus RB, Hendriks JCM, Messele T, Borghans J, Miedema F, Wolday D, Coutinho RA, and Dukers NHTM. Low CD4 T Cell Counts before HIV-1 Seroconversion Do Not Affect Disease Progression in Ethiopian Factory Workers, The Journal of Infectious Diseases, **192**, 739 - 748 (2005).

76. Meshkani MR, Billard L. Empirical Bays estimators for a finite Markov Chain. Biometrika, **79**:1, 195 - 193 (1992).

77. Mestern M. Distributed analysis of Markov chains. Thesis (M.Sc. (Computer Science)), University of Cape Town (1998).

78. http://www.microsoft.com/windows/windows-7/

79. Ncube N. Markov operators on Schatten classes. Thesis (M.Sc.), University of South Africa (2000).

80. Nel EM. Estimating the pen trajectories of static handwritten scripts using hidden Markov models. Thesis (PhD), University of Stellenbosch (2005).

81. Niezen G. The optimization of gesture recognition techniques for resource-constrained devices. Thesis (M.Eng.(Computer Engineering)), University of Pretoria, 2008.

82. Ocana-Riola R. Non-homogenous Markov Processes for Biomedical Data Analysis. Biometrical Journal, **47**:3, p 369 - 376 (2005).

83. Ochola RO. Investigation of strain rate sensitivity of polymer matrix composites. Thesis (PhD. (Materials Engineering)), University of Cape Town (2004).

84. Oosthuizen DR. Markov models for mobile radio data communication systems. Proefskrif (M.Ing. (Elek.)), Randse Afrikaanse Universiteit (1990).

85. Parks KP, Bentley LR, Crowe AS. Capturing Geological Realism in Stochastic Simulations of Rock Systems with Markov Statistics and Simulated Annealing. Journal of Sedimentary Research; **70**:4, 803 - 813 (2000).

86. Pérez-Ocón R, Ruiz-Castro JE, Gámiz-Pérez ML. Markov models with lognormal transition rates in the analysis of survival times. Test, **9**:2, 353 - 370 (2000).

87. Personal communication on the 17th of March 2010 with Prof Tienie Stander; Honorary Professor of the North West University in the School of Pharmacy, and 2008/2009 president of the South African chapter of the International Society of Pharmacoeconomics and Outcomes Research.

88. Piccart-Gebhart MJ et al. Trastuzumab after Adjuvant Chemotherapy in HER2-Positive Breast Cancer. The New England Journal of Medicine, **353**:16, 1659 - 1672 (2005).

89. Prinsloo GJ. Phoneme class recognition and automatic syllabification with a phonological based hidden Markov model. Thesis (M.Ing.), University of Stellenbosch (1988).

90. Purnell DW. Discriminative and Bayesian techniques for hidden Markov model speech recognition systems. Thesis (D.Phil. (Engineering)), University of Pretoria (2001).

91. Richard A, Richardson S, Maccario J. A Three-State Markov Model of Plasmodium falciparum Parasitemia. Mathematical Biosciences, **117**, 283 - 300 (1993).

92. R Development Core Team (2009). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

93. Roodt AE. Algorithm for a stack search HMM word spotter. Thesis (M.Ing.), University of Stellenbosch (1995).

94. Rosenthal JS. A First Look at Rigorous Probability Theory, Second Edition. World Scientific Publishing Co. Pte. Ltd. (2006).

95. Sahin AD, Sen Z. First-order Markov chain approach to wind speed modeling. Journal of Wind Engineering and Industrial Aerodynamics, **89**, 263 - 269 (2001).

96. Satten GA, Longini IM. Markov chains with measurement error: Estimating the 'true' course of a marker of the progression of human immunodeficiency virus disease. Applied Statistics – Journal of the Royal Statistical Society Series C, **45**:3, 275 - 295 (1996).

97. Schwardt L. Efficient mixed-order hidden Markov model inference. Dissertation (PhD), University of Stellenbosch (2007).

98. Sharples LD. Use of the Gibbs sampler to estimate transition rates between grades of coronary disease following cardiac transplantation. Statistics in Medicine, **12**, 1155 - 1169 (1993).

99. Sindle C. Handwritten signature verification using hidden Markov models. Thesis (MScIng), University of Stellenbosch (2003).

100. Singer B and Spilerman S. The Representation of Social Processes by Markov Models. The American Journal of Sociology, **82**:1, 1 - 54 (1976).

101. Strydom D. Recognition of speech in additive convolutional noise. Thesis (M.Ing.), University of Stellenbosch (1995).

102. Swarts F. Markov characterization of fading channels. Proefskrif (M.Ing.(Elek.)), Randse Afrikaanse Universiteit (1991).

103. Sweeting MJ et al. Estimated progression rates in three United Kingdom hepatitis C cohorts differed according to method of recruitment. Journal of Clinical Epidemiology, **59**:2, 144 - 152 (2006).

104. Tuma NB, Brandon N, Hannan MT. Approaches to the censoring problem in analysis of event histories. Sociological Methodology, **10**, 209 - 240 (1979).

105. Tuma NB, Hannan MT, Groeneveld LP. Dynamic analysis of Event histories. The American Journal of Sociology, **84**:4, 820 - 854 (1979).

106. Van der Berg G. Hidden Markov models for tool wear monitoring in turning operations. Thesis (M.Eng.(Mechanical Engineering)), University of Pretoria (2004).

107. Van der Merwe CJ. Phonetic alignment of speech using continuous density hidden Markov models. Study project (M. Elec. Eng.), University of Stellenbosch (1991).

108. Van der Merwe HJ. Bird song recognition with Hidden Markov Models. Thesis (MScIng), University of Stellenbosch (2008).

109. Van Heerden CJ. Phoneme duration modelling for speaker verification. Thesis (M.Eng.(Computer Engineering), University of Pretoria, 2008.

110. Van Heerden RP. Hidden Markov models for robust recognition of vehicle licence plates. Thesis (M.Eng.)(Computer.), University of Pretoria (2002).

111. Van Rooyen. Sustained learning for a speech recognition system. Thesis (M.Ing.), University of Stellenbosch (1994).

112. Waardenburg T. The automatic recognition of stop consonants. Dissertation (PhD), University of Stellenbosch (1994).

113. Walker AJ. A fast flexible digital hardware implemented Markov data source. Thesis (PhD.), University of the Witwatersrand (1978).

114. Welton NJ, Ades AE. Estimation of Markov Chain Transition Probabilities and Rates from Fully and Partially Observed Data Uncertainty Propagation, Evidence Synthesis, and Model Calibration. Medical Decision Making, **25**:6, 633 - 645 (2005).

115. Wessels T. Hidden Markov models for on-line signature verification. Thesis (M.Sc), University of Stellenbosch (2002).

116. Zhou W. An experimental evaluation of Markov channel models. M.Ing (Elektriese en Elektroniese Ingenieurswese), Randse Afrikaanse Universiteit (1998).