POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Ingegneria Matematica

A BAYESIAN ANALYSIS OF NON-DECREASING LONGITUDINAL DATA USING BIRTH-DEATH PROCESSES

Relatore: Prof.ssa Alessandra Guglielmi
Correlatore: Prof. Michael Peter Wiper

Tesi di Laurea di:
Andrea Bassi
Matr. 781633

Anno Accademico 2012-2013

Ringraziamenti

Vorrei dapprima ringraziare le persone che mi hanno aiutato nella stesura di questa tesi, da sei mesi a questa parte.

Ringrazio il professor Michael Wiper, che mi ha seguito e aiutato in modo costante, dandomi sempre utili suggerimenti e confortandomi quando non ottenevo i risultati sperati.

Un sincero grazie alla professoressa Alessandra Guglielmi, la quale mi ha dapprima trasmesso l'interesse per la statistica bayesiana e successivamente mi ha seguito con dedizione aiutandomi nella stesura di questa tesi, con tanti suggerimenti e correzioni utili.

Ci tengo anche a ringraziare Enza, per il suo aiuto con la lingua inglese e per le sue correzioni.

Dopo aver ringraziato chi mi ha aiutato nello specifico, vorrei dedicare questo lavoro a tutte le persone che mi sono state vicine e che mi hanno sempre incoraggiato durante questi cinque lunghi anni di università.

Grazie Mamma, per essermi stata sempre vicina, dimostrandomi il tuo affetto in ogni modo possibile.

Ringrazio mio Papà, per avermi sempre incoraggiato e aiutato nel corso della mia vita, trasmettendomi dei valori di cui ho fatto tesoro e spero di portare sempre con me.

Un ringraziamento speciale alle mie nonne, Liliana e Wanda, perché quando veniva meno la concentrazione o la motivazione per raggiungere un obiettivo, ho sempre pensato a voi e ho subito ritrovato le forze per andare avanti, onorato dal sapere quanto bene mi volete e quanto siete orgogliose di me.

Grazie anche a chi non c'è più, mi riferisco ai miei nonni Giovanni e Tullio, che hanno potuto vedere solo l'inizio di questo mio percorso, ma spero che da lassù festeggino con me questo traguardo importante.

Subito dopo i familiari, un affettuoso pensiero non può che andare agli amici, coloro che hanno riempito di colore e gioia questi miei anni da studente.

Grazie a Giovanni, Marco, Federico e Francesco, perché ho sempre potuto

contare sulla vostra più sincera amicizia e mi sono sempre sentito a mio agio e sereno in vostra compagnia.

Ringrazio Alessandro, Carlo e Michele, per avermi fatto apprezzare ancora di più questi cinque anni di università grazie alla vostra compagnia.

Grazie a Elena, Luciano e Massimiliano, che sono entrati nella mia vita da poco ma mi hanno subito conquistato e hanno reso la mia esperienza universitaria all'estero molto più ricca e divertente.

Grazie ad Alessandro, compagno di mille avventure solastiche e non, senza di te di certo mi sarei goduto meno tutto questo percorso.

Grazie a Marta, per gli innumerevoli aiuti e consigli che mi hai dato, rivelandoti una splendida persona su cui poter sempre contare.

Grazie a Paolo, o forse dovrei dire al professor Camassa, per essere stato il primo a trasmettermi la passione per la matematica e a scorgere in me quelle potenzialità che hai sempre sostenuto e incoraggiato, dando il via a quello che poi sarebbe diventato il mio futuro universitario e non solo.

Infine, grazie a tutti quelli che per evidenti motivi di spazio non rientrano in questa (già lunga) lista, ma che hanno un posto di riguardo nel mio cuore.

Andrea

Riassunto

In questo lavoro verrà presentata una classe di modelli bayesiani per l'analisi di dati longitudinali non decrescenti. In particolare, in questo elaborato verranno analizzati dati relativi alla crescita di bambini affetti da leucemia linfoblastica acuta (LLA). Il data set raccoglie dati relativi all'altezza di ogni paziente misurata dal pediatra in istanti di tempo con cadenza media attorno ai 6 mesi. Oltre all'altezza, per ogni paziente è segnalato il tipo di trattamento a cui è stato sottoposto, dividendo i pazienti in tre categorie. Studi medici hanno ipotizzato un rallentamento del processo di crescita per i pazienti sottoposti a sedute di radioterapia (gruppi 2 e 3); uno degli obiettivi di questo lavoro è la verifica di tale ipotesi tramite opportune analisi statistiche.

Si tratta di rappresentare le curve di crescita, per ogni paziente, con un vettore $(Y_t,\ t \in \{t_1, ..., t_n\})$. Si assume che Y_t, l'altezza di ogni paziente all'istante t, sia pari ad un fattore di scala J, costante nel tempo, moltiplicato per l'integrale, rispetto al tempo, di un processo stocastico costante a tratti; il tempo non è lineare, ma è riparametrizzato con una funzione di *time-scaling*.

Assumeremo che la funzione di *time-scaling* sia nota, oppure sia parametrizzata linearmente, e in questo caso i parametri che la rappresentano verrano stimati.

I parametri da cui dipende la verosimiglianza dei dati sono vari: alcuni sono intesi come variabili latenti, come il numero o la frequenza dei salti del processo di nascita e morte; altri invece sono parametri di interesse (sui quali faremo inferenza), come il fattore di scala J e la funzione di *time-scaling*. Entrambi i modelli verranno poi estesi al caso di dati raggruppati, per trattare il problema medico preso in esame. Seguendo l'approccio bayesiano verrà assegnata una distribuzione a priori per il vettore di parametri.

Valuteremo le inferenze bayesiane per tale classe di modelli, utilizzando due diversi tipi di algoritmi per il calcolo della distribuzione finale: un algoritmo *Gibbs sampler* e uno di tipo ABC (Approximate Bayesian Computation). In particolare, porremo attenzione al confronto dei parametri per gruppo (cioè la diversa terapia per i pazienti), verificando che tale differenza viene confermata dalle nostre stime.

Abstract

In this work a class of Bayesian models will be introduced to analyze non-decreasing longitudinal data. In this thesis we will analyze data related to the growth of children affected by acute lymphoblastic leukemia (ALL). The data set collects information about measurements on height for every patient, taken at diagnosis approximately every 6 months. Besides the height, for each patient is reported the type of treatment to which it was subjected, dividing the data in three categories. Previous studies on the effects of cranial radiation on height suggested that radiation (used in groups 2 and 3) contributed to decreased expected height. One of the goals of this work is to verify this hypothesis with proper statistical analyses.

We are representing the growth curves, for each patient, with a vector $(Y_t,\ t \in \{t_1, ..., t_n\})$. We assume that Y_t, the height of each patient at time t, is equal to a scale factor J, time independent, times the integral of a piecewise constant stochastic process; time is not linear, but is reparametrized with a *time-scaling* function. We will assume that this functions is known, or linearly parametrized, and in this case the parameters of which is composed will be estimated by statistical analyses.

The conditional likelihood depends on various parameters, some of them are latent variables, such as the number or frequency of jumps of the birth-death process; on the other hand we have parameters of interest (on which we are interested to make inference), like the scale factor J and the *time-scaling* function. Both models will be extended to the case of grouped data, to treat the specific medical problem. Following the Bayesian approach we will assign a prior distribution to the vector of parameters.

We will compute the Bayesian inferences for this class of models, using two different kinds of algorithms to evaluate the final distribution: a standard Gibbs sampler and an ABC (Approximate Bayesian Computation) algorithm. In particular, we will focus attention on comparing parameters between groups (each of them represents a different therapy used on patients) and we will verify the hypothesis of different growth trends for the three groups.

Contents

List of Figures

List of Tables

Introduzione

In questo lavoro verrà presentata una classe di modelli bayesiani per l'analisi di dati longitudinali non decrescenti. I dati longitudinali sono un caso particolare di misure ripetute, nel caso in cui le variabili di interesse sono state misurate per ogni soggetto o unità sperimentale in diversi istanti di tempo successivi o in condizioni 'sperimentali' diverse. In particolare, siamo interessati a studiare dei dati longitudinali non decrescenti, che sono solitamente chiamati *curve di crescita*.

In questo elaborato verranno analizzati dati relativi alla crescita di bambini affetti da leucemia linfoblastica acuta (LLA). Questi dati sono stati resi disponibili dal Dana Farber Cancer Institute (Boston, USA) e sono stati inviati dalla prof.ssa Maria Durbán, una degli autori di Durbán et al. (2004), al prof. Michael Peter Wiper. In uno studio clinico sono state raccolte informazioni sulla crescita di 618 pazienti durante un periodo di tempo che va da Novembre 1987 a Dicembre 1995. Il data set raccoglie dati relativi all'altezza di ogni paziente misurata dal pediatra in istanti di tempo con cadenza media attorno ai 6 mesi. Oltre all'altezza, per ogni paziente è segnalato il tipo di trattamento a cui è stato sottoposto, dividendo di fatto i pazienti in tre categorie: quelli curati senza radioterapia, quelli sottoposti a un trattamento di radiazioni considerato standard e quelli a cui sono state applicate radiazioni iper-frazionate. Studi medici precedenti hanno ipotizzato un rallentamento del processo di crescita per i pazienti sottoposti a sedute di radioterapia; uno degli obiettivi di questo lavoro è la verifica di tale ipotesi tramite opportune analisi statistiche.

I modelli considerati sono due e presentano una simile struttura della distribuzione condizionale dei dati. Entrambi i modelli condividono alcuni parametri e, ovviamente, il modo di interpretare i dati, che sono curve di crescita. La curva di crescita di ogni paziente è rappresentata da un vettore $(Y_t,\ t \in \{t_1, ..., t_n\})$. Si assume che Y_t sia proporzionale, tramite un fattore di scala aleatorio J, all'integrale $\int_0^{G(t)} U_s\, ds$. Qui $\{U_s,\ s \in (0, G(t))\}$ è un processo di nascita e morte con tasso λ, con spazio degli stati $\mathcal{S} = \{0, 1, 2, ...k\},\ k \in \mathbb{N}$. $G(t)$ invece è una funzione continua non-decrescente,

che chiameremo di *time scaling*. Dunque ad ogni istante di tempo, assumeremo che l'altezza di ogni paziente sia pari ad uno stesso fattore di scala J (che non dipende dal tempo) per l'integrale rispetto al tempo di un processo stocastico costante a tratti; il tempo non è lineare, ma è riparametrizzato con la funzione $G(t)$.

In particolare, il primo modello che abbiamo considerato assume che la funzione di *time scaling* sia nota, mentre nel secondo la funzione viene parametrizzata e i parametri che la compongono, incogniti, saranno oggetto di analisi statistiche.

I parametri da cui dipende la verosimiglianza dei dati sono vari: alcuni sono intesi come variabili latenti, come il numero o la frequenza dei salti del processo di nascita e morte; altri invece sono parametri di interesse (sui quali andremo a fare inferenza), come il fattore di scala J e la funzione di *time-scaling*. Entrambi i modelli verranno poi estesi al caso di dati raggruppati, per trattare il problema medico preso in esame. Seguendo l'approccio bayesiano verrà assegnata una distribuzione a priori per il vettore dei parametri. Le inferenze sono costruite sulla base della distribuzione finale, cioè la distribuzione condizionale dei parametri, date le osservazioni.

In sintesi, lo scopo di questa tesi è verificare se un tale modello sia ragionevole per dati che sono curve di crescita, e in particolare per il data set relativo alla crescita di bambini affetti da LLA. Ci siamo concentrati sulla stima bayesiana dei parametri J e $G(t)$. Particolare attenzione è stata posta nella stima dei parametri all'interno dei tre gruppi in cui i dati sono stati suddivisi. Prima di applicare i modelli proposti ai dati clinici a disposizione è stata 'testata' la validità dei modelli con un'analisi effettuata su un data set costruito *ad hoc*, simulando i dati dalla densità condizionale del modello bayesiano scelto, e fissando i parametri del modello. Dopo questa prima fase, in cui i modelli sono stati leggermente modificati (per adattarsi al problema preso in esame), è stata condotta l'analisi bayesiana sui dati 'reali', confrontando le stime a posteriori dei parametri dei citati gruppi. Per ogni modello le stime a posteriori sono state ottenute attraverso due metodi: il metodo MCMC standard e il metodo ABC (Approximate Bayesian Computation).

Entrambi gli algoritmi sono stati implementati utilizzando il software `R` (R Development Core Team, 2009). Sono stati scritti i codici per effettuare le simulazioni di tipo Markov Chain Monte Carlo (MCMC), cioè simulazioni di traiettorie da una catena markoviana, aperiodica ed irriducibile, la cui distribuzione limite è la posterior del modello considerato. La preparazione dei dati e l'analisi degli output sono state effettuate utilizzando il pacchetto `mcmc`. Sempre in `R` sono stati scritti i codici per le simulazioni di tipo ABC, utilizzando gli stessi modelli bayesiani.

Nel Capitolo 1 vengono richiamate anzitutto le nozioni probabilistiche riguardanti le catene markoviane, che sono alla base della modellizzazione delle nostre curve di crescita. In seguito viene introdotto il modello di nascita e morte per costruire $Y_t = J \int_0^{G(t)} U_s ds$ e vengono calcolati i momenti del processo. Infine si presenta il concetto di divisione aleatoria di un intervallo e viene calcolata la densità di probabilità della statistica di ordine k-esimo su un campione n-dimensionale sull'intervallo, secondo l'approccio descritto in David and Nagaraja (2003); la conoscenza della distribuzione di questa quantità sarà utile in seguito, quando verrà costruita la legge condizionale delle osservazioni dati i parametri, del modello bayesiano proposto qui.

Nel Capitolo 2 sono descritti i due modelli bayesiani utilizzati nel corso del lavoro; cioè la legge condizionale dei dati, dato il vettore dei parametri, e la prior scelta per i parametri stessi. Nello stesso capitolo è stato incluso un paragrafo che illustra gli algoritmi utilizzati per ottenere le stime a posteriori dei parametri; in particolare si fa un richiamo generale sui metodi MCMC spiegando nel dettaglio l'algoritmo utilizzato nel lavoro: il *Gibbs sampler* con passi di *Metropolis-Hastings*. Questo genere di algoritmo è basato sul campionamento di ciascuno dei parametri di interesse θ_i a partire dalle loro distribuzioni *full conditional*, cioè dalle distribuzioni a posteriori condizionali $\mathcal{L}(\theta_i|\theta_{-i})$ dati tutti gli altri parametri e le osservazioni. Pertanto nel capitolo sono calcolate le *full conditional* dei due modelli proposti e vengono presentati gli pseudocodici utili a capire gli algoritmi utilizzati.

Il Capitolo 3 presenta una tecnica diversa per ottenere le distribuzioni a posteriori dei parametri, il cosiddetto metodo ABC (Approximate Bayesian Computation). In questo capitolo vengono introdotti alcuni algoritmi alla base di questo metodo, seguendo le trattazioni di Marjoram et al. (2003) e Marin et al. (2011) sull'argomento. Infine vengono illustrati gli algoritmi ABC utilizzati nele nostre applicazioni.

Il Capitolo 4 contiene tutte le applicazioni: si descrive il procedimento per simulare i dati dalla distribuzione condizionale delle osservazioni quando i parametri sono fissati e vengono descritte alcune simulazioni di prova e successivamente è presente un'ampia descrizione dei data set utilizzati per le inferenze, il primo proveniente dai dati relativi alla crescita dei pazienti e il secondo simulato usando la procedura appena descritta. Poi è riportata l'analisi bayesiana dei parametri nei due modelli, utilizzando sia i dati simulati che quelli reali. L'ultima parte del Capitolo 4 riporta le stime a posteriori dei parametri utilizzando gli stessi modelli ma con approccio ABC.

Questa tesi sviluppa alcuni aspetti dei modelli presentati nel Capitolo 4 della tesi di dottorato di Ana Paula Palacios [vedi Palacios (2012)]. In par-

ticolare il modello di partenza era stato applicato a un problema di crescita di una colonia batterica. In questo lavoro il modello di Palacios (2012) è stato modificato ed esteso introducendo nuovi parametri, come quelli che costituiscono la funzione di *time-scaling*; la legge condizionale dei dati, condizionatamente ai parametri, così risulta diversa da quella di Palacios (2012) per due aspetti: qui la funzione di *time-scaling* è incognita (ma è stata considerata lineare a tratti) e i dati sono raggrupati (e quindi il nostro è un modello gerarchico).

Introduction

In this work a class of Bayesian models will be introduced to analyze non-decreasing longitudinal data. Longitudinal data are a specific case of repeated measurements, in which the variables are measured for each subject or sperimental unit in different subsequent times. In particular, we are interested in studying non-decreasing longitudinal data, which are often named *growth curves*.

In this thesis we will analyze data related to the growth of children affected by acute lymphoblastic leukemia (ALL). This data was collected at Dana Farber Cancer Institute (Boston, USA) and was sent to prof. Michael Peter Wiper from prof. Maria Durbán, one of the authors of Durbán et al. (2004). In one of the clinical trials carried out a total of 618 children were treated between November 1987 and December 1995 with three different central nervous system therapies: intrathecal therapy alone (no radiation), intrathecal therapy with conventional cranial radiation, and intrathecal therapy with twice daily radiation. Measurements on height and weight were taken at diagnosis and approximately every 6 months thereafter. Previous studies on the effects of cranial radiation on height suggested that radiation contributed to decreased expected height, since cranial radiation has been associated with the development of growth hormone deficiency. One of the goals of this work is to verify this hypothesis with proper statistical analyses.

We will introduce two models, with a similar structure of the conditional distribution of the data. Both models share some parameters and the way of interpreting data, that are growth curves. The growth curve for each patient is represented by a vector $(Y_t,\ t \in \{t_1, ..., t_n\})$. We assume that Y_t is proportional, through a random scale factor J, to the integral $\int_0^{G(t)} U_s\, ds$. Here $\{U_s,\ s \in (0, G(t))\}$ is a birth-death process of rate λ, with state space $\mathcal{S} = \{0, 1, 2, ...k\},\ k \in \mathbb{N}$. $G(t)$ is a non-decreasing continuous function, which we will call *time-scaling function*.

In particular, the first model we considered assumes that the *time-scaling function* is known, while in the second one the function is parametrized and the parameters of which is composed of will be estimated by statistical

analyses.

The conditional likelihood depends on various parameters: some of them are latent variables, such as the number or frequency of jumps of the birth-death process; on the other hand we have parameters of interest (on which we are interested to make inference), like the scale factor J and the *time-scaling function*. Both models will be extended to the case of grouped data, to treat the specific medical problem. Following the Bayesian approach we will assign a prior distribution for the vector of parameters, which includes J, some scalar parameters which represent $G(t)$ and some latent variables. Inferences are built basing on the final distribution, namely the conditional distribution of the parameters, given the observations.

To sum up, the aim of this thesis is to verify if this certain model is reasonable for growth curve data and, in particular, for our data set related to the growth of children affected by ALL. We computed the Bayesian estimate of the parameters J and $G(t)$, focusing on how parameters change within the three groups in which the data set is divided. Before applying the proposed models to the real data set, the robustness of the models has been tested through an analysis driven with a simulated data set, built by sampling data from the conditional likelihood of the data (fixing the parameters). After this first step, in which our models were properly modified (to adapt to this specific study case), we led to the analysis on real data, comparing the posterior estimates of the parameters of every group. For both models the posterior estimates were obtained by two methods: the classic Bayesian approach and the ABC method (Approximate Bayesian Computation).

All the models were implemented using the `R` software (R Development Core Team, 2009). We wrote the codes to run every Monte Carlo Markov Chain simulation, that are simulation of trajectories from a Markovian chain, aperiodic and irreducible, which limiting distribution is the posterior of the selected model. Preparation of the data and output analysis were driven using the `mcmc` package. The codes for ABC simulations were also written in `R`.

In Chapter 1 we start with a brief theoretical review about Markov chains, which are the starting point for modeling our growth curves. Afterwards we introduce the birth-death process for the data and the moments of the process are computed. In the end we present the topic of the random division of an interval and we compute the probability density of the k-th order statistic over a n-dimensional sample on a time interval, following the approach described in David and Nagaraja (2003); the knowledge of this distribution will be useful when we will define the conditional distribution of the data, given the parameters, for the two Bayesian models.

In Chapter 2 we describe in detail the two Bayesian models used throughout this work; namely the conditional distribution of the data, given the parameters vector, and the prior distribution of the parameters. In the same chapter there is a section which introduces the algorithm used to obtain the posterior estimates of the parameters, in particular, there is a general review of MCMC methods focusing on the algorithm used in this work: the *Metropolis-Hastings within Gibbs sampler*. This type of algorithm is based on sampling each parameter θ_i from its *full conditional* distribution, that is the conditional posterior distribution $\mathcal{L}(\theta_i|\theta_{-i})$ given all the parameter and the observations. Therefore, in the chapter we computed the *full conditionals* of the two proposed models and wrote all the pseudocodes in order to understand the algorithms used.

In Chapter 3 we introduce a different technique to get the posterior distribution of the parameters, the so called ABC method (Approximate Bayesian Computation). In this chapter we show some basic ABC algorithms, following Marjoram et al. (2003) and Marin et al. (2011). Then we move to explain the ABC approach regarding our study case, describing the algorithms used in our applications.

Chapter 4 contains all the applications: we first describe the procedure to simulate data from the conditional likelihood. Afterwards there is a detailed description of the data sets used for inference purposes, the first one from clinical data and the second simulated using the just mentioned algorithm. In addition, the posterior estimates of the parameters are reported, for each model using both simulated and real data. The last part of Chapter 4 contains the posterior estimates of the same parameters using ABC approach instead of the ordinary MCMC.

This thesis develops some aspects of the models presented in Chapter 4 of the doctoral thesis by Ana Paula Palacios [see Palacios (2012)]. In particular, the starting model was applied to a bacterial growth problem. In this work, the model of Palacios (2012) has been modified and extended by introducing new parameters, such as the ones which constitute the *time-scaling function*; the conditional distribution of the data, given the parameters, is different from the one described in Palacios (2012) for two aspects: first because the *time-scaling* function is unknown (we assumed it piecewise linear), secondly, in this case data are grouped (and so we are dealing with a hierarchical model).

Chapter 1

Probabilistic background

In this chapter we will introduce all the concepts needed to fully understand the upcoming statistical analysis.

We will start with a review of Markov chains theory, focusing on the most important properties that are exploited throughout the following chapters. Then, we will introduce the growth curve model and compute its moment generating function. After all we will lay the foundations for computing the model likelihood by describing how to obtain the distribution of the k-th order statistic over a sample of n random values on a finite interval.

1.1 Markov chains theory

Here we will describe what is a Markov chain. In particular, we will define both discrete-time and continuous-time Markov chains and show their basic properties.

Definition 1. A *random process* is a collection of random variables indexed by some set $\mathcal{T}$, taking values in some (countable) set $\mathcal{S}$.

- $\mathcal{T}$ is the index set, usually time.
- Each $i \in \mathcal{S}$ is called *state* and $\mathcal{S}$ is called the *state-space*.

We classify random processes according to both the index set (discrete or continuous) and the state space (finite, countable or uncountable/continuous).

Definition 2. A random process is called a *Markov process* if, conditional on the current state of the process, its future is independent of its past.

More formally, $\{X_t\}$ is Markovian if has the following property:

$$\mathbb{P}(X_{t_n} = i_n \,|\, X_{t_{n-1}} = i_{n-1}, ..., X_{t_1} = i_1) = \mathbb{P}(X_{t_n} = i_n \,|\, X_{t_{n-1}} = i_{n-1}) \quad (1.1)$$

for all finite sequences of times $t_1 < ... < t_n \in \mathcal{T}$ and of states $i_1, ..., i_n \in \mathcal{S}$.

Definition 3. A Markov chain X_t is said to be *time-homogeneous* if

$$\mathbb{P}(X_{s+t} = j \,|\, X_s = i)$$

is independent of s. When this holds, putting $s = 0$ gives

$$\mathbb{P}(X_{s+t} = j \,|\, X_s = i) = \mathbb{P}(X_t = j \,|\, X_0 = i).$$

1.1.1 Discrete-time Markov chains

If we assume that the process changes from one state i to another state j only at fixed time epochs ($n = 1, 2, 3, ...$) we can say that $\{X_n\}_{n\geq 0}$ is a *discrete-time Markov chain* or briefly DTMC.

To define correctly a DTMC we need to introduce the initial distribution ν. We say that $\nu = (\nu_i,\ i \in \mathcal{S})$ is a *measure* on $\mathcal{S}$ if $0 \leq \nu_i \leq \infty$ for all $i \in \mathcal{S}$. If in addition the *total mass* $\sum_{i\in\mathcal{S}} \nu_i$ equals 1, then we call ν a *distribution*. Then ν defines a distribution, the *distribution of* X. We think of X as modelling a random state which takes the value i with probability ν_i.

We can write the probabilities of transition from one state to another

$$p_{ij} = \mathbb{P}(X_{n+1} = j \,|\, X_n = i), \quad i, j \in \mathcal{S},\ \forall n \in \mathbb{N}.$$

By collecting all the probabilities in a matrix we obtain the so called *one-step transition matrix* $\mathbf{P} = (p_{ij},\ i, j \in \mathcal{S})$, this matrix has the property that every row $(p_{ij},\ j \in \mathcal{S})$ is a distribution.

We shall now formalize the rules for a Markov chain by a definition in terms of the corresponding matrix $\mathbf{P}$.

Definition 4. We say that $\{X_n\}_{n\geq 0}$ is a *discrete-time Markov chain* with *initial distribution* ν and *transition matrix* $\mathbf{P}$ if

1. X_0 has distribution ν;

2. for $n \geq 0$, conditional on $X_n = i, X_{n+1}$ has distribution $(p_{ij},\ j \in \mathcal{S})$ and is independent of $X_0, ..., X_{n-1}$

Class structure

It is sometimes possible to break a Markov chain into smaller pieces, each of which is relatively easy to understand,and which together give an understanding of the whole. This is done by identifying the communicating classes of the chain.

We say that i *leads to* j and write $i \to j$ if

$$\mathbb{P}(X_n = j \,|\, X_0 = i, \quad \text{for some } n \geq 0) > 0.$$

We say i *communicates with* j and write $i \leftrightarrow j$ if both $i \to j$ and $j \to i$.

The relation $\leftrightarrow$ satisfies the conditions for an equivalence relation on $\mathcal{S}$, and thus partitions the state space into *communicating classes*. A chain or transition matrix $\mathbf{P}$ where the state space consists of a single communicating class is called *irreducible*.

Classification of states

Let $\{X_n\}_{n\geq 0}$ be a Markov chain with transition matrix $\mathbf{P}$. We say that a state i is emphrecurrent if

$$\mathbb{P}(X_n = i \text{ for infinitely many } n) = 1.$$

We say that i is *transient* if

$$\mathbb{P}(X_n = i \text{ for infinitely many } n) = 0.$$

Thus a recurrent state is one to which you keep coming back and a transient state is one which you eventually leave for ever.

Recurrence and transience are class propertier, for instance if two states are in the same communicating class then they are recurrent/transient together. We therefore speak of recurrent or transient classes.

Invariant distributions

Remember that a measure ν is any row vector $(\nu_i,\ i \in \mathcal{S})$ with non-negative etries. We say ν is *invariant* (but also the terms *stationary* or *equilibrium* are used) if

$$\nu\mathbf{P} = \nu.$$

The follow results explain the terms stationary and equilibrium.

Theorem 1. Let $\{X_n\}_{n\geq 0}$ be Markov$(\nu, \mathbf{P})$ and suppose that ν is invariant for $\mathbf{P}$. Then $\{X_{m+n}\}_{n\geq 0}$ is also Markov$(\nu, \mathbf{P})$.

Theorem 2. Let $\mathcal{S}$ be finite and let $p_{ij}^{(n)}$ be the probability to move from state i to state j in n steps, namely $p_{ij}^{(n)} = \mathbb{P}(X_{m+n} = j \,|\, X_m = i)$. Suppose for some $i \in \mathcal{S}$ that

$$p_{ij}^{(n)} \to \pi_j \text{ as } n \to \infty \text{ for all } j \in \mathcal{S}.$$

Then $\pi = (\pi_j,\ j \in \mathcal{S})$ is an invariant distribution.

1.1.2 Continuous-time Markov chains

In most of the real applications we cannot consider time as an equally spaced grid of points, we have to leave the assumption that events happen only at prescribed time points and deal with a continuous time.

It may be slightly more difficult to deal with continuous-time Markov chains (CTMC) because there is no real equivalent to the one-step transition matrix from which one can calculate all quantities of interest.

The study of CTMCs is based on the *transition function*. If we denote by $p_{ij}(t)$ the probability of a process starting in state i being in state j after elapsed time t, then we call $P(t) = (p_{ij}(t),\ i,j \in \mathcal{S}, t > 0)$ the transition function of that process. $P(t)$ is difficult to write down in all but the simplest of situations. However it is proved that there exist quantities $q_{ij},\ i,j \in \mathcal{S}$ satisfying

$$q_{ij} = p'_{ij}(0^+) = \begin{cases} \lim_{t\to 0^+} \frac{p_{ij}(t)}{t}, & i \neq j, \\ \\ \lim_{t\to 0^+} \frac{1-p_{ii}(t)}{t}, & i = j. \end{cases}$$

We call the matrix $\mathbf{Q} = (q_{ij},\ i,j \in \mathcal{S})$ the *q-matrix of the process* and we can interpret it as follows:

- for $i \neq j,\ q_{ij} \in [0,\infty)$ is the instantaneous rate the process moves from state i to state j,
- $q_i = -q_{ii} \in [0,\infty]$ is the rate at which the process leaves state i.

We also have that $\sum_{j\neq i} q_{ij} \leq q_i$.

When we formulate a model, it is $\mathbf{Q}$ that we can write down, so the problem now is to recover $P(\cdot)$ from $\mathbf{Q} = P'(0)$. If we have a q-matrix $\mathbf{Q}$, then the transition function $P(t)$ must satisfy the so called *Kolmogorov backward equations*

$$P'(t) = \mathbf{Q}P(t), \quad t > 0,$$

and may or may not satisfy the *forward Kolmogorov equations*

$$P'(t) = P(t)\mathbf{Q}, \quad t > 0,$$

with the initial condition $P(0) = I$.

There is always one such transition function, but there may also be infinitely many such functions, so $\mathbf{Q}$ does not necessarily describe the whole process.

Interpreting the Q-matrix

Suppose $X_0 = i$, then we have that the holding time H_i in state i is exponentially distributed with parameter q_i.

$$\mathbb{P}(H_i \leq t) = 1 - e^{-q_i t}, \quad t \geq 0.$$

That is why we are speaking about *transition rates*, the diagonal entry $-q_{ii}$ represents the expected time spent in state i before jumping to another state.

The process could jump into any other communicating state, the probability that the process jumps to state j is q_{ij}/q_i.

Limiting behaviour

As with discrete-time chains, the class structure is important in determining what tools are useful for analysing the long term behaviour of the process. The notions of recurrence/transience and irreducibility are the same as in the discrete case.

If the state space is irreducible and positve recurrent, the limiting distribution is the unique (up to constant multiples) solution $\pi = (\pi_i,\ i \in \mathcal{S})$ such that

$$\pi\mathbf{Q} = \mathbf{0},$$

where $\mathbf{0}$ is a vector of zeros. If $\sum_i \pi_i < \infty$, then π can be normalised to give a probability distribution which is the limiting distribution. (If π is not summable then there is no proper limiting distribution.)

Poisson processes

Poisson processes are some of the simplest examples of continuous-time Markov chains. Such proceses are the natural probabilistic models for any uncoordinated stream of discrete events in continuous time.

Given a right-continuous process $\{X_t\}_{t\geq 0}$ we can obtain its *jump times* $J_0, J_1, ...$ and their respective *holding times* $S_1, S_2, ...$ These quantities are obtained by

$$J_0 = 0, \quad J_{n+1} = \inf\{t \geq J_n : X_t \neq X_{J_n}\}$$

for $n = 0, 1, ...$ where $\inf \emptyset = \infty$, and, for $n = 1, 2, ...,$

$$S_n = \begin{cases} J_n - J_{n-1} & \text{if } J_{n-1} < \infty \\ \infty & \text{otherwise.} \end{cases}$$

In other words, jump times mark the times in which the process jumps from any state to another, while holding times are the length of intervals between two subsequent jumps.

The discrete-time process $\{Y_n\}_{n\geq 0}$ given by $Y_n = X_{J_n}$ is called the *jump chain* of $\{X_t\}_{t\geq 0}$.

Now we can give a definition of the Poisson process in terms of jump chain and holding times. A right-continuous process $\{X_t\}_{t\geq 0}$ with values in $\{0, 1, 2, ...\}$ is a *Poisson process of rate* λ $(0 < \lambda < \infty)$ if its holding times $S_1, S_2, ...$ are independent exponential random variables of parameter λ and its jump chain is given by $Y_n = n$. The associated q-matrix is given by

$$\mathbf{Q} = \begin{pmatrix} -\lambda & \lambda & & \\ & -\lambda & \lambda & \\ & & \ddots & \ddots \\ & & & \end{pmatrix}.$$

A simple way to construct a Poisson process of rate λ is to take a sequence $S_1, S_2, ...$ of independent exponential random variables of parameter λ, to set $J_0 = 0, J_n = S_1 + ... + S_n$ and then set

$$X_n = n \quad \text{if} \quad J_n \leq t < J_{n+1}.$$

Another important result is a property that we will exploit in the next chapters: if $\{X_t\}_{t\geq 0}$ is a Poisson process of rate λ, then it has stationary independent increments, and, for each t, X_t has Poisson distribution of parameter λt.

An *increment* over any interval $(s, t]$ is $X_t - X_s$. We say that $\{X_t\}_{t\geq 0}$ has *stationary* increments if the distribution of $X_{s+t} - X_s$ depends only on $t \geq 0$. We say that $\{X_t\}_{t\geq 0}$ has *independent* increments if its increments over any finite collection of disjoint invervals are independent.

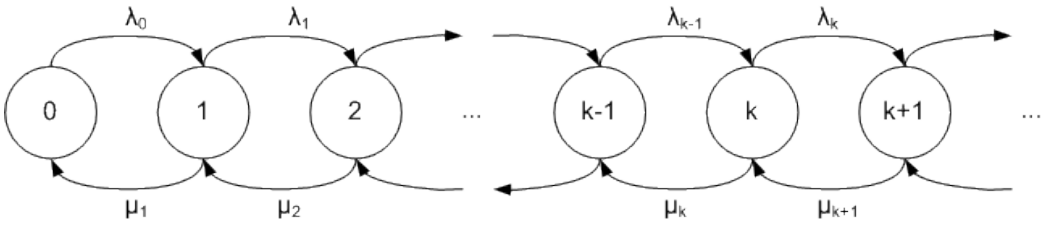


Figure 1.1: Birth-death process.

1.2 A birth-death model for non decreasing curves

Consider a birth-death process (BDP), $\{U_t : t \geq 0\}$, that is a special case of a continuous-time Markov process where the state transition are of only two types: an increase of the state variable by one and a decrease of the same amount (births and deaths).

Then if our process U_t is in the state i, after an exponential amount of time it will move to one of the neighbouring states $i \to i-1$ or $i \to i+1$.

Once the space state $\mathcal{S} = \{0, 1, 2, ..., k\}$ is defined we can uniquely determine the process giving the generator matrix, $\mathbf{Q}$, and the initial distribution of the process, $\boldsymbol{\nu}_0$.

This is the typical generator matrix for a BDP:

$$\mathbf{Q} = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \cdots & 0 \\ \mu & -(\lambda+\mu) & \lambda & 0 & \cdots & 0 \\ 0 & \mu & -(\lambda+\mu) & \lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \mu & -(\lambda+\mu) & \lambda \\ 0 & 0 & \cdots & 0 & -\mu & \mu \end{pmatrix}$$

which is a tri-diagonal matrix, where the parameters $\lambda, \mu > 0$ are, respectively, the insantaneous birth and death rates. This means that each time the process enters in the state i the amount of time it spends before making a transition to a neighbour state is a random variable, say T, exponentially distributed with parameter $\xi = \lambda + \mu$. Then the process enters in the state $i+1$ with probability $\frac{\lambda}{(\lambda+\mu)}$ or in the state $i-1$ with probability $\frac{\mu}{(\lambda+\mu)}$. Obviously, in the edge states (0 and k), the only chain transition possible is to the right/left neighbour.

The growth processes that we are interested to study are nearly always represented with a continuous curve. Therefore, the most natural and simple way to obtain a continuous (and non decreasing) curve from a piecewise

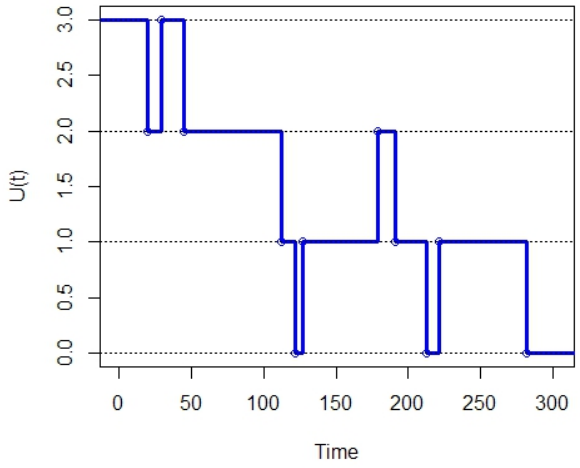


Figure 1.2: One possible realization of the U_t process.

constant process is by doing the cumulative integral. We define a continuous state process, $\{V_t : t \geq 0\}$ such that

$$V_t = J \int_0^t U_s ds \ , \tag{1.2}$$

where J is a positive constant.

The stochastic process V_t defined in (1.2) is called *subordinator* and represents the basis for our growth curve model. The last step in modeling the curve consists of a deterministic time change, we finally define our stochastic growth process $\{Y_t : t \geq 0\}$ as

$$Y_t = V_{G(t)} \tag{1.3}$$

where V_t is as defined in (1.2) and $G(t)$ is a continuous non decreasing function.

The time transformation is important to get a process similar to the observed data, different kinds of growth curves present typical shapes that differ on the context (e.g. bacteria growth, children growth, crack size tests, pollutant concentration in a source of water etc).

It is not recommended to model specific sets of data with the raw process V_t, because it presents a very generic shape. Then, especially when we are facing at data presenting different growth phases, we need to apply the deterministic time change. This is done in order to improve the model, because

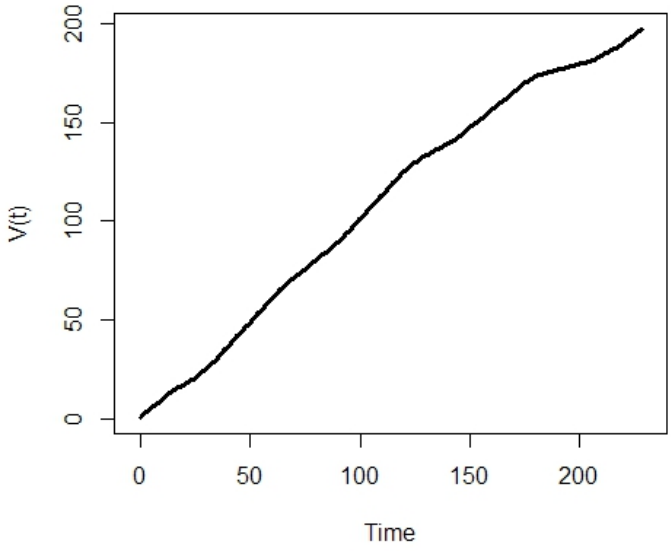


Figure 1.3: One possible realization of the V_t process.

passing from V_t to $Y_t = V_{G(t)}$ we will get a curve more similar to the function $G(t)$.

We will also see how the time change function should be chosen, because there is a relationship between the function $G(t)$ and the mean of the process. Keeping in mind this we can 'tune' our model, by changing the G function, to make it more accurate.

1.3 Moments of the growth process

In this section we compute the moments of the process $\{Y_t\}$, introduced in (1.3). As we use a deterministic time change, it is sufficient to compute the expected trajectory of the subordinator, $m_V(t) = E[V_t]$, and then to apply the time transformation to it.

The expected trajectory can be computed as

$$\begin{aligned} E[V_t] &= E\left[J\int_0^t U_s(\omega)ds\right] \\ &= \int_\Omega J\int_0^t U_s(\omega)ds\,dP(\omega) \\ &= J\int_0^t\int_\Omega U_s(\omega)ds\,dP(\omega) \\ &= J\int_0^t E[U_s]ds\ , \end{aligned} \tag{1.4}$$

that is Fubini's theorem to change the order of the integration.

Now we left only the computation of the expected trajectory of the process U_t, we consider only the case in which U_t is in stationary state (irreducible chain with all states positive recurrent).
In this case the Markov chain has a stationary distribution given by

$$\Pi_i = \frac{\left(\frac{\lambda}{\mu}\right)^i}{\sum_{h=0}^k \left(\frac{\lambda}{\mu}\right)^h} \qquad \text{for } i = 0, ..., k. \tag{1.5}$$

Then $\eta := E[U_t] = \sum_{i=0}^k i\,\Pi_i$ and therefore

$$\begin{aligned} E[V_t] &= J\int_0^t E[U_s]ds \\ &= JE[U_t]t \\ &= J\eta\ t\ . \end{aligned}$$

Finally, to obtain the mean trajectory we just have to change the time,

$$E[Y_t] = J\eta\ G(t)\ . \tag{1.6}$$

For what concerns the higher order moments we can compute the Laplace transform of the subordinator, $\mathcal{L}\{f(V_t)\} = f_V^*(s) = E[e^{-sV_t}]$. Then it is straightforward to obtain $f_Y^*(s)$ by applying the time change, and finally we can compute the n-th moment of the process exploiting a property of the Laplace transform:

$$E[X^n] = (-1)^n \left.\frac{d^n f_X^*(s)}{ds^n}\right|_{s=0} \tag{1.7}$$

where X is a random variable and $f_X^*(s)$ its Laplace transform. The Laplace transform can be computed as

$$\begin{aligned} f_V^*(s) &= E[e^{-sV_t}] = E[e^{-sJ\int_0^t U_r dr}] \\ &= \sum_{i=0}^{k} \Pi_i \, e^{-sJ\int_0^t i\, dr} \\ &= \Pi_0 + \Pi_1 \, e^{-sJ\int_0^t 1\, dr} + ... + \Pi_k \, e^{-sJ\int_0^t k\, dr} \\ &= \Pi_0 + \Pi_1 \, e^{-sJt} + \Pi_2 \, e^{-2sJt} + ... + \Pi_k \, e^{-ksJt} \end{aligned} \tag{1.8}$$

where Π_i are the stationary probabilities computed in (1.5).

To get the Laplace transform of $\{Y_t\}$ we just have to apply the already mentioned time change to the transform of the subordinator $\{V_t\}$.

$$f_Y^*(s) = f_V^*(s)\Big|_{t=G(t)} = \Pi_0 + \Pi_1 \, e^{-sJG(t)} + \Pi_2 \, e^{-2sJG(t)} + ... + \Pi_k \, e^{-ksJG(t)} \ . \tag{1.9}$$

Once we have obtained this expression we have everything we need to calculate any moment of the process for every parameter choice (remember that the probabilities of the stationary distribution depend on the birth/death rates), just by applying the (1.7).

1.4 Random division of an interval

In this section we present a topic that will be useful later while computing the likelihood function of the model. Then, we need to introduce the distribution of the k-th order statistic over a sample, following David and Nagaraja (2003).

Suppose that n points are dropped at random on the unit interval (0,1). The ordered distances of these points from the origin are denoted by $u_{(i)}$ $(i = 1, 2, ..., n)$ and let $w_i = u_{(i)} - u_{(i-1)}$ $(u_{(0)} = 0)$ be the interval between them. Then the random variables $U_{(1)}, U_{(2)}, ..., U(n)$ are distributed as n order statistics from a uniform $\mathcal{U}(0, 1)$ parent, that is, with joint pdf equal to $n!$ over the simplex $0 \leq u_{(1)} \leq u_{(2)} \leq ... \leq u_{(n)} \leq 1$. Correspondingly, the pdf of the w_i is

$$f(w_1, w_2, ..., w_n) = n! \quad w_i \geq 0, \quad \sum_{j=1}^{n} w_j \leq 1 \ . \tag{1.10}$$

The distribution is completely symmetrical in the w_i. Indeed, if we define

$$w_{n+1} = 1 - \sum_{j=1}^{n} w_j \tag{1.11}$$

we have the (degenerate) joint probability density function ($j = 1, 2, ..., n, n+1$)

$$f(w_1, w_2, ..., w_n, w_{n+1}) = n! \quad w_i \geq 0, \quad \sum_{j=1}^{n+1} w_j = 1 \tag{1.12}$$

which is still symmetrical in all w_j. It follows that the joint distribution of any k of the W_j $(k = 1, 2, ..., n)$ is the same as that of the first k, and in particular that the distribution of the sum of any k of the W_j is that of

$$U_{(k)} = W_1 + W_2 + ... + W_k \tag{1.13}$$

namely

$$f_{U_{(k)}}(u) = \frac{1}{B(k, n+1-k)} u^{k-1}(1-u)^{n-k} \quad 0 \leq u \leq 1 \ . \tag{1.14}$$

The W_js are commonly referred to as *spacings*.

The random division of the interval may in fact originate from a Poisson process, such as our problem, with events occurring in some interval of time. Then, the distribution of the k-th order statistic, $U_{(k)}$, in the interval $[0, T]$ is a scaled beta distribution, $\mathcal{B}(k, n+1-k)$:

$$f_{U_{(k)}}(u) = \frac{1}{B(k, n+1-k)} \frac{u^{k-1}(T-u)^{n-k}}{T^n} \quad 0 \leq u \leq T \ . \tag{1.15}$$

Chapter 2

Bayesian inference

In this chapter we will present two Bayesian models for the growth curve process: a basic approach and a more sophisticated one. Then, we will focus on the problem of estimating the model parameters from the data, for this purpose, we are interested in computing the posterior distribution of the parameters given the data. Since the direct computation will not be possible, we will introduce a very popular computational method, the MCMC algorithms, to obtain samples from the target distribution.

2.1 Preliminary assumptions

Assume that we observe the heights, $q_1 < ... < q_n$ of a child at a sequence of time points, say $0 < t_1 < ... < t_n$.

In the model defined by (1.3), the likelihood function is analytically unavailable, but for the case of two state in the Markov process U_t, we can find an explicit expression for the likelihood when conditioning on the initial state and the number of jumps in successive time intervals. Therefore, from now on, we shall consider just the process with two states, $\mathcal{S} = \{0, 1\}$, and with equal jump rate for birth and death ($\lambda = \mu$).

Our simplified model can be written as:

$$\mathbf{Q} = \begin{pmatrix} -\lambda & \lambda \\ \lambda & -\lambda \end{pmatrix} \qquad \mathcal{S} = \{0, 1\}$$

$$Y_t = V_{G(t)} = J \int_0^{G(t)} U_s \, ds \qquad \lambda, J > 0 \tag{2.1}$$

where $\mathbf{Q}$ is the generator matrix of the process U_t and $\mathcal{S}$ is the state space.

First we have to transform the data from heights at time t to increments of height during the (transformed) time interval.

$$\begin{aligned} y_1 &= q_1 \\ y_i &= q_i - q_{i-1}\ , \qquad i = 2, ..., n \\ g_1 &= G(t_1) \\ g_i &= G(t_i) - G(t_{i-1})\ , \quad i = 2, ..., n \end{aligned} \tag{2.2}$$

Then, for $i = 1, ..., n$, y_i is J times the total (transformed) time spent in state 1 in interval i, so that we can write

$$y_i = Jg_{i1}, \text{ where } g_i = G(t_i) - G(t_{i-1}) = g_{i0} + g_{i1}$$

g_i is the size of the i-th transformed time interval and g_{i0} (g_{i1}) is the total transformed time spent in state 0 (1) in interval i. This allows for the computation of the conditional likelihood.

2.2 Model A, a basic approach

In this first model we suppose that G is a known function. Let the initial state at the start of the first time interval be s_1 and let m_i represent the number of jumps made by the process in the i-th time interval for $i = 1, ..., n$.

2.2.1 Conditional distribution of data

From now on we consider the increments of height (defined in (2.2)) as our data, we are also considering the time intervals after the transformation made by using $G(t)$.

Then, the likelihood function is:

$$f(\mathbf{y}|J, \lambda, s_1, m_1, \ldots, m_n) = \frac{1}{J^n} \prod_{i=1}^{n} f(g_{i1}|J, \lambda, s_i = \text{mod}(s_{i-1} + m_{i-1}, 2), m_i) \tag{2.3}$$

where $\text{mod}(a, b)$ represents a modulo b and $g_{i1} = y_i/J$ is the time spent in state 1 in interval i. The densities of the g_{i1}s are conditionally independent given the state at the start of interval i and the number of state transitions in the interval.

Now consider two cases: when m_i is odd and when m_i is even. Consider now the different time intervals in each state.

- If m_i is odd, the process spends half of the time intervals in state 1 and the remainder in state 0 and therefore, the distribution of the sum of $(m_i + 1)/2$ intervals is equal to the distribution of the order statistic $U_{((m_i+1)/2)}$ as defined in (1.15), that is:

$$f(g_{i1}|m_i, \lambda) = \frac{1}{B(\frac{m_i+1}{2}, \frac{m_i+1}{2})} \frac{g_{i1}^{(m_i+1)/2-1} g_{i0}^{(m_i+1)/2-1}}{g_i^{m_i}} \tag{2.4}$$

- If m_i is even, then the process spends $m_i/2 + 1$ time intervals in the state at the start of the interval and $m_i/2$ in the other state. Therefore, from (1.15),

$$f(g_{i1}|\lambda, s_i, m_i) = \frac{1}{B(\frac{m_i}{2} + s_i, \frac{m_i}{2} + 1 - s_i)} \frac{g_{i1}^{m_i/2+s_i-1} g_{i0}^{m_i/2-s_i}}{g_i^{m_i}} \tag{2.5}$$

  where $g_{i0} = g_i - g_{i1}$.

We can notice that the distribution of the time spent in the state 1 in the i'th interval is a scaled Beta distribution (with range $[0, g_i]$) with parameters that depends on the latent variables m_i and s_i.

Multiple observations

Here above we computed the likelihood function for a single observation, that is one vector of increasing values $\mathbf{y}$ which represents the sampling of a growth curve at fixed time epochs. When we are facing at multiple observations we can easily extend our thoughts and consider the likelihood for the whole set of curves.

Let $\mathbf{Y}$ be a $m \times n$ matrix, where m is the number of different growth curves and n is the length of the time grid where the curves were evaluated. We can arrange our data set in this fashion:

$$\mathbf{Y} = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_m \end{pmatrix} = \begin{pmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & & \vdots \\ y_{m1} & \cdots & y_{mn} \end{pmatrix},$$

where each row of the matrix represents a single growth process.

We can consider the reailzations of the different processes all independent, thus it is straightforward to write down the distribution of the whole data set $\mathbf{Y}$ conditioning on the parameters, the only difference is that in this case we have to take into account a greater number of m_j's and s_1's because of the greater number of processes considered.

$$
\begin{aligned}
f(\mathbf{Y}|J,\lambda,\mathbf{s}_1,\mathbf{M}) &= \prod_{j=1}^{m} f(\mathbf{y}_j|J,\lambda,\mathbf{s}_{1j},\mathbf{m}_j) \\
&= \prod_{j=1}^{m} \left\{ \frac{1}{J^n} \prod_{i=1}^{n} f(g_{ji_1}|J,\lambda,s_{1ji},m_{ji}) \right\} \\
&= \frac{1}{J^{mn}} \prod_{i,j} f(g_{ji_1}|J,\lambda,s_{1ji},m_{ji}), \qquad (2.6)
\end{aligned}
$$

where $\mathbf{M}$ is a matrix (same dimensionality as $\mathbf{Y}$) which includes the number of jumps for every interval and for every curve, namely $\{\mathbf{M}\}_{ji} = m_{ji} =$ number of jumps in the i'th interval of the j'th curve.

2.2.2 Choosing the prior distribution

We are assuming that the time scale function $G(t)$ is known and fixed. This function represents a sort of average behaviour of the growth process, if we have specific information about the process (e.g. an opinion of an expert) we can use a specific function. Otherwise $G(t)$ can be estimated from the data itself.

The other parameters that remain unknown are:

- J, jump size
- λ, jump rate
- s_1, a binary value representing the initial state of the process
- $\mathbf{m} = (m_1, m_2, ..., m_n)'$, a $n-$dimensional vector containing the number of jumps per time interval

It would be easier for computational reasons to assume that the parameters are all independent from each other, to write down a prior distribution that is the product of all the marginal priors. This is just partially applicable in our study case, because we have to keep in mind that there is a strong dependence between every m_i. Apart from this correlation, we can assume that the initial state and the jump size are independent from all the other variables. Therefore, the prior distribution can be written in this form:

$$
\pi(J,\lambda,s_1,\mathbf{m}) = \pi(J)\,\pi(s_1)\,\pi(\lambda,\mathbf{m}) = \pi(J)\,\pi(s_1)\,\pi(\mathbf{m}|\lambda)\,\pi(\lambda) \qquad (2.7)
$$

- $J \sim \text{Gamma}(\eta,\tau)$

- $\lambda \sim \text{Gamma}(\alpha, \beta)$
- $s_1 \sim \text{Be}(p)$
- $\pi(\mathbf{m}|\lambda) = \prod_{i=1}^{n} \pi(m_i|\lambda) = \prod_{i=1}^{n} \text{Poi}(\lambda g_i)$

where $g_i = G(t_i) - G(t_{i-1})$.

The natural choice for the prior of s_1 is a Bernoulli distribution, and in particular we fixed $p = 1/2$ because we do not have any information about a more likely starting point for our process.

The jump size J has to be positive, so we chose a Gamma distribution for its prior. We can tune the parameters η and τ depending on our prior knowledge about the jump size. In most of the cases we will choose a non informative prior because the lack of information, this can be easily done by choosing small values (less than the unit) both for η and τ.

As we have already explained, the m_i are independent and identically distributed. They follow a Poisson distirbution with a mean value that is the product of the jump rate and the length of the time interval (after the transformation).

For what concerns λ, we chose a Gamma distribution because this kind of prior is partially conjugated. This means that, as we will see, the conditional posterior distribution of λ will also belong to the Gamma family. This fact will speed up any simulation algorithm because we will only have to update the parameters of the distribution instead of calculating a new one.

2.3 Model A.2, grouped data

As we will see later, in our practical case, we have to work with a grouped data set. Every growth curve has a label indicating the group which the data belongs to. In this section we will write down the likelihood function and the prior in the case of a known time scale function. We will see how the parameters shall be modified to introduce the variability between groups.

2.3.1 Conditional distribution of data

The main difference between this model and model A is that now we are assuming three different jump sizes, one for each group. From now on we are

assuming that the number of groups is K and, for simplicity, that in every group we have N curves, made by n observation.

Our parameter set is $(\mathbf{J}, \lambda, \mathbf{M}, \mathbf{s_1})$, where

- $\mathbf{J} = (J_1, ..., J_K)$ is a K-dimensional vector containing the jump sizes for each group.
- λ is the jump rate, equal for every group.
- $\mathbf{M} = \{\mathbf{M_1}, ..., \mathbf{M_K}\}$ is a set of K matrixes, each composed of N rows and n columns. Every matrix $\mathbf{M_j}$ contains all the number of jumps for every curve belonging to the j-th group. Every row of the matrix represents a single curve, while every column represents a time interval.
- $\mathbf{s_1} = (\mathbf{s_1}_1, ..., \mathbf{s_1}_K)$ is a collection of K vectors of length equal to N. $\mathbf{s_1}_j$ contains the information about the initial state for every curve in group j.

Assuming the independence both within the groups and between them it is straightforward to compute the likelihood. It is just the product of the likelihood of every growth curve data, paying attention to which group the data belongs to.

$$\begin{aligned} f(\mathbf{Y}|\mathbf{J}, \lambda, \mathbf{M}, \mathbf{s_1}) &= f(\mathbf{Y}_1|\mathbf{J}, \lambda, \mathbf{M}, \mathbf{s_1})...f(\mathbf{Y}_K|\mathbf{J}, \lambda, \mathbf{M}, \mathbf{s_1}) \\ &= f(\mathbf{Y}_1|J_1, \lambda, \mathbf{M}_1, \mathbf{s_1}_1)...f(\mathbf{Y}_K|J_K, \lambda, \mathbf{M}_K, \mathbf{s_1}_K) \\ &= \prod_{j=1}^{K} f(\mathbf{Y}_j|J_j, \lambda, \mathbf{M}_j, \mathbf{s_1}_j) \end{aligned} \tag{2.8}$$

where $\mathbf{Y} = \{\mathbf{Y}_1, ..., \mathbf{Y}_K\}$ is the whole dataset, divided into groups. The same division was also made for the latent variables, the number of jumps and the initial state. The three functions are the already shown likelihood in the case of multiple observations (see (2.6)), with the respective parameters for each group.

2.3.2 Choosing the prior distribution

We are doing the same assumptions, in terms of independence, as we did for model A. The only difference is that in this case we have to manage a higher number of parameters, thus the resulting prior distribution will have a higher

dimensionality but in the specific will remain very similar to the one chosen in model A.

$$\begin{aligned}\pi(\mathbf{J},\lambda,\mathbf{M},\mathbf{s_1}) &= \pi(\mathbf{J})\pi(\mathbf{M}|\lambda)\pi(\lambda)\pi(\mathbf{s_1}) \\ &= \pi(\lambda)\prod_{j=1}^{K}\pi(J_j)\pi(\mathbf{M}_j|\lambda)\pi(\mathbf{s_1}_j). \qquad (2.9)\end{aligned}$$

This means that a priori the parameters are independent even at group level, for example the jump size for group i is independent not only from λ or s_1 of the same group, also from all the other jump sizes. We are keeping the same distribution for the parameters, for the J_j-s is a Gamma, for $\mathbf{M}_j$ a product of independent Poisson and for $\mathbf{s_1}_j$ a Bernoulli. In conclusion,

- $J_j \sim \text{Gamma}(\eta_j, \tau_j)$, with $j = 1, ..., K$.
- $\lambda \sim \text{Gamma}(\alpha, \beta)$.
- $\{\mathbf{M}_j\}_{pq}|\lambda \sim \text{Poi}(\lambda g_q)$, for all $p = 1, .., N$.
- $(\mathbf{s_1}_j)_p \sim \text{Be}(\theta)$, for all $p = 1, ..., N$.

2.4 Model B, parametrizing G

In this section we propose another model. Up to now, we have always considered the $G(t)$ function fixed, now we are assuming that the function has unknown parameters. The original model now has been modified, by eliminating the jump rate λ (fixing it equal to 1), and extended by parametrizing the function $G(t)$.

2.4.1 The choice of the time scale function

We have already discussed the strong relationship between the time scale function $G(t)$ and the mean of the process, in fact the growth process $\{Y_t\}$ has a mean trajectory that is proportional to the G function.

Then choosing a good time scaling is important to make the model adapt to the specific data, shaping the curves generated by the model as the available ones.

The data are sequences of observations over time, so we need to interpolate the growth curves for each individual and then taking the mean curve.

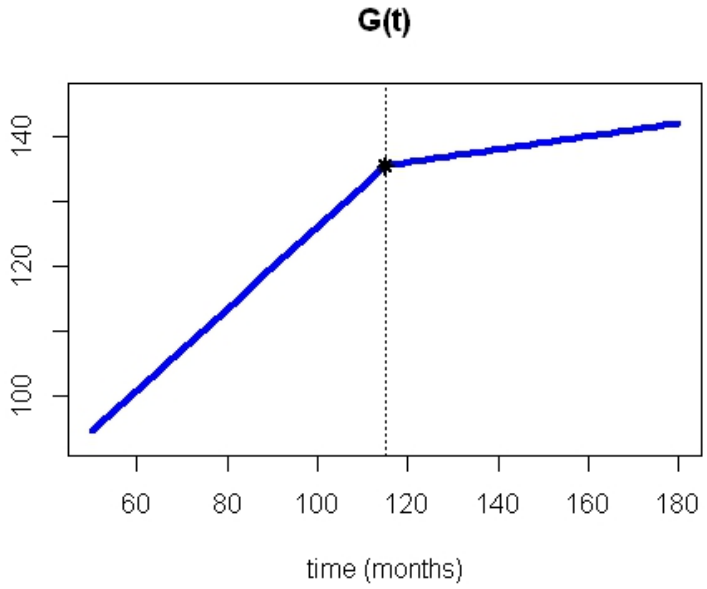


Figure 2.1: An example of a time scale function.

Observing the three groups of data (growth curves of children treated with different therapies) we realized that the growth trend is quite linear, but it seems that there is a decreasing of the growth speed after the first ten years of life (we will deeply analyze the data set in the last chapter).

So for a first approximation we chose to find a parametric curve as our $G(t)$, the simple shape of the curve is a broken line with one corner point, that is the simplest way to represent the general trend of the children growth.

$$G(t) = \begin{cases} a + bt, & \text{for } t \leq t^* \\ c + dt, & \text{for } t > t^*, \end{cases} \tag{2.10}$$

where the corner point is $t^* = \frac{c-a}{b-d}$.

2.4.2 Conditional distribution of data

Given the parameters of the time scale function, the likelihood for the data of this model is exactly the same as the one in the previous model.

$$f(\mathbf{y}|J, G, s_1, M_1 = m_1, \ldots, M_n = m_n) = \frac{1}{J^n} \prod_{i=1}^{n} f(g_{i1}|J, G, s_i = \text{mod}(s_{i-1}+m_{i-1}, 2), m_i) \tag{2.11}$$

With G we mean the knowledge of all the parameters that define the function, namely (a, b, c, d, t^*). Also remember that $g_{i1} = y_1/J$.

As in the previous model we have to distinguish two cases to explicit each of the n independent terms in the productory.

- If m_i is odd,

$$f(g_{i1}|m_i, G) = \frac{1}{B(\frac{m_i+1}{2}, \frac{m_i+1}{2})} \frac{g_{i1}^{(m_i+1)/2-1} g_{i0}^{(m_i+1)/2-1}}{g_i^{m_i}} \tag{2.12}$$

- If m_i is even,

$$f(g_{i1}|s_i, m_i, G) = \frac{1}{B(\frac{m_i}{2} + s_i, \frac{m_i}{2} + 1 - s_i)} \frac{g_{i1}^{m_i/2+s_i-1} g_{i0}^{m_i/2-s_i}}{g_i^{m_i}} \tag{2.13}$$

  where $g_{i0} = g_i - g_{i1}$.

Remember that, as we did for model A (see (2.6)), we can easily extend the likelihood in the case of multiple observations.

2.4.3 Choosing the prior distribution

To complete the Bayesian model we have to choose a suitable prior distribution for the set of parameters, that are

- J, jump size
- $\mathbf{G}$, time scale function
- s_1, a binary value representing the initial state of the process
- $\mathbf{m} = (m_1, m_2, ..., m_n)'$, a n-dimensional vector containing the number of jumps per time interval

The time scale function is of this form (see figure 2.1)

$$G(t) = \begin{cases} a + bt, & \text{for } t \leq t^* \\ c + dt, & \text{for } t > t^* \end{cases}$$

To avoid overparametrization issues we choose to take one of the five parameters (a, b, c, d, t^*) as known given the other four.

The time scale function is proportional to the mean of the growth process, in our study case we are assuming a mean that increases linearly with

two different speeds, the turning point is represented by t^* while the slope parameters (b, d) give us information about the two speeds. Therefore, these three variables must be involved in the analysis.

We choose to put a prior on the set (a, b, d, t^*) and later compute $c = a + (b - d)t^*$.

We are assuming that some parameters are independent from the others, namely we consider two blocks that are independent from each other: (J, s_1) and $(\mathbf{G}, \mathbf{m})$.

Let's consider the first block: we don't have reasons to think that the jump size J infects in any way the initial state s_1 neither vice versa.

On the contrary we can not assume the independency within the second block because the number of jumps in each interval is distributed as a Poisson with a parameter that is the length of the transformed time interval. $m_i|\mathbf{G} \sim \text{Poi}(g_i), \quad g_i = G(t_i) - G(t_{i-1})$.
Since the transformation depends on the scale function we conclude that $\mathbf{m}$ and $\mathbf{G}$ are dependent, thus we are setting the prior distribution in this way:

$$\pi(J, s_1, \mathbf{m}, \mathbf{G}) = \pi(J, s_1)\,\pi(\mathbf{m}, \mathbf{G}) = \pi(J)\,\pi(s_1)\,\pi(\mathbf{m}|\mathbf{G})\,\pi(\mathbf{G}) \tag{2.14}$$

- $J \sim \text{Gamma}(\eta, \tau)$
- $s_1 \sim \text{Be}(p)$
- $\pi(\mathbf{m}|\mathbf{G}) = \prod_{i=1}^{n} \pi(m_i|\mathbf{G}) = \prod_{i=1}^{n} \text{Poi}(g_i)$
- $\pi(\mathbf{G}) = \pi(a)\,\pi(b)\,\pi(d)\,\pi(t^*) = \prod_j \text{Gamma}(\alpha_j, \beta_j)$

For the priors of s_1 and J we are keeping the same assumptions as in model A.

Thus, the natural choice for the prior of s_1 is a Bernoulli distribution with mean equal to $1/2$ and for the jump size J is a Gamma distribution. We have already discussed about the choice of the distribution and the possibility to tune the parameters η and τ in order to add our prior knowledge about the jump size to the model.

A similar choice was made for the four parameters that make up $\mathbf{G}$, every of them must be strictly positive because of their meaning. The two slope parameters, b and d, since the time scale has to be non decreasing, as the whole process. It is obvious that also the changing slope time t^* has to be positive. Finally we have that $a > 0$ because a is related to the mean of the process at time zero ($G(0) = a$), thus it has no sense to assume that the growth process starts with a negative mean value.

2.5 Model B.2, grouped data

In this section we extend model B for grouped data, the idea is the same as for model A.2 but fixing λ and parametrizing $G(t)$.

2.5.1 Conditional distribution of data

We are assuming a data composed of K groups of N curves. Each curve is sampled in an equally spaced time grid, thus every curve is made by n observations.

In this case the parameter set is $(J, \mathbf{G}, \mathbf{M}, \mathbf{s_1})$, where

- J is the jump size, equal for each group.
- $\mathbf{G} = \{G_1, ..., G_K\}$ is the set of K time scale functions (all following (2.10)), one for each group. Every function is parametrized in the same way, $G_j = (a_j, b_j, d_j, {t_j}^*)$, where $c_j = a_j + (b_j - d_j){t_j}^*$.
- $\mathbf{M} = \{\mathbf{M_1}, ..., \mathbf{M_K}\}$ is the same set of K matrices defined in model A.2.
- $\mathbf{s_1} = (\mathbf{s_{11}}, ..., \mathbf{s_{1}}_K)$ is a collection of K vectors of length equal to N. $\mathbf{s_{1}}_j$ contains the information about the initial state for every curve in group j.

We kept the same jump size for all groups because, as we will see in the following examples, it is sufficient (and easier) to assume three different time scale functions to study the variability of the data between groups.

We are assuming (as we did for model A.2) the independence of all these parameters, both within the groups and between them. Then, we can obtain the likelihood function as

$$\begin{aligned} f(\mathbf{Y}|\mathbf{J}, \mathbf{G}, \mathbf{M}, \mathbf{s_1}) &= f(\mathbf{Y}_1|\mathbf{J}, \mathbf{G}, \mathbf{M}, \mathbf{s_1})...f(\mathbf{Y}_K|\mathbf{J}, \mathbf{G}, \mathbf{M}, \mathbf{s_1}) \\ &= f(\mathbf{Y}_1|J, G_1, \mathbf{M}_1, \mathbf{s_{11}})...f(\mathbf{Y}_K|J, G_K, \mathbf{M}_K, \mathbf{s_{1}}_K) \\ &= \prod_{j=1}^{K} f(\mathbf{Y}_j|J, G_j, \mathbf{M}_j, \mathbf{s_{1}}_j) \end{aligned} \tag{2.15}$$

where $\mathbf{Y} = \{\mathbf{Y}_1, ..., \mathbf{Y}_K\}$ is the whole dataset, divided into groups. The same division was also made for the latent variables, the number of jumps and the initial state. The three functions are the already shown likelihood in the case of multiple observations (see (2.6)), with the respective parameters for each group.

2.5.2 Choosing the prior distribution

In this case of grouped data we deal with multiple time scale functions. We can keep all the assumptions did for model B and adding a higher dimensional prior for $\mathbf{G}$.

$$\begin{aligned} \pi(J, \mathbf{G}, \mathbf{M}, \mathbf{s_1}) &= \pi(J)\pi(\mathbf{M}|\mathbf{G})\pi(\mathbf{G})\pi(\mathbf{s_1}) \\ &= \pi(J)\prod_{j=1}^{K}\pi(\mathbf{M}_j|G_j)\pi(G_j)\pi(\mathbf{s_1}_j). \end{aligned} \tag{2.16}$$

All the parameters are independent between groups. Within every group the only dependency that we have to take into account is the one between G_j and $\mathbf{M}_j$.

- $J \sim \text{Gamma}(\eta, \tau)$.
- $\pi(G_j) = \pi(a_j)\,\pi(b_j)\,\pi(d_j)\,\pi(t_j{}^*)$, all the four parameters are independent from each other, for all $j = 1, ..., K$.
  - $a_j \sim \text{Gamma}(\alpha_a, \beta_a)$.
  - $b_j \sim \text{Gamma}(\alpha_b, \beta_b)$.
  - $d_j \sim \text{Gamma}(\alpha_d, \beta_d)$.
  - ${t^*}_j \sim \text{Gamma}(\alpha_{t^*}, \beta_{t^*})$.
- $\{\mathbf{M}_j\}_{pq}|\lambda \sim \text{Poi}(\lambda g_q)$, for all $p = 1, .., N$.
- $(\mathbf{s_1}_j)_p \sim \text{Be}(\theta)$, for all $p = 1, ..., N$.

2.6 Markov chain Monte Carlo (MCMC) algorithms

In this section we will explain why we needed to use MCMC methods and how they work.

In particular, we will focus our attention to two algorithms: the Metropolis-Hastings and the Gibbs sampler.

2.6.1 Monte Carlo principle

Our principal aim from now on will be to get as more information as we can of the parameters of the model (that in this section we will indicate with $\boldsymbol{\theta}$) given a certain data set, say $\mathbf{y}$. In other words, we are interested to the random variable $\boldsymbol{\theta}|\mathbf{y} = y$.

Here a simple idea can be helpful: *anything we want to know about a random variable can be learned by sampling many times from its density.* This is known as the *Monte Carlo principle*.

Unfortunately, in our study case we do not know the density of $\boldsymbol{\theta}|\mathbf{y}$, in most of the practical cases the posterior distribution is analytically unavailable because of a too high computational task.

The key mathematical tool that can avoid this problem is a Markov chain, we will see how it is possible to generate Markov chains that have a given target density, $\pi(\boldsymbol{\theta}|\mathbf{y})$, as the Markov chain's invariant density. The important fact to keep in mind here is that the Monte Carlo principle applies even when the samples are not independent, but form a Markov chain.

2.6.2 Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm defines a set of 'jumping rules' that generate a Markov chain on the support of $\pi(\boldsymbol{\theta}|\mathbf{y})$. At the start of iteration k, we have $\boldsymbol{\theta}^{(k-1)}$ and we make the transition to $\boldsymbol{\theta}^{(k)}$ as follows:

1. `sample` $\boldsymbol{\theta}^*$ `from a 'proposal' distribution` $Q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(k-1)})$
2. $$r \leftarrow \frac{\pi(\boldsymbol{\theta}^*|\mathbf{y})\, Q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(k-1)})}{\pi(\boldsymbol{\theta}^{(k-1)}|\mathbf{y})\, Q(\boldsymbol{\theta}^{(k-1)}|\boldsymbol{\theta}^*)} \tag{2.17}$$
3. $\alpha \leftarrow min(r, 1)$
4. `sample` $U \sim \mathcal{U}(0, 1)$
5. **if** $U \leq \alpha$ **then**
6. $\quad \boldsymbol{\theta}^{(k)} \leftarrow \boldsymbol{\theta}^*$
7. `else`
8. $\quad \boldsymbol{\theta}^{(k)} \leftarrow \boldsymbol{\theta}^{(k-1)}$
9. **end if**

The quantity r is called *acceptance ratio*, it means that if $r > 1$ then the algorithm makes the transition $\boldsymbol{\theta}^{(k)} \leftarrow \boldsymbol{\theta}^*$ with probability 1; otherwise we make the transition with probability r (remember that $\Pr(r \leq U) = r$ if $U \sim \mathcal{U}(0,1)$).

The overall performance of the algorithm is strongly influenced by the choice of the candidate density Q, that is the key to the algorithm.

Theoretical results prove that using this scheme we can generate a Markov chain that has the posterior density $\pi(\boldsymbol{\theta}|\mathbf{y})$ as its invariant distribution.

2.6.3 Gibbs sampling

When $\boldsymbol{\theta}$ is high dimensional, as if often the case in many statistical models, sampling from the posterior density $\pi(\boldsymbol{\theta}|\mathbf{y})$ could be too hard even for the Metropolis-Hastings algorithm, because finding a good joint proposal density could be complicated.

In these cases we avoid the problem of the high dimensionality by sampling from a series of inter-related, easier and lower-dimensional densities instead that from the posterior density. We will see how to do this such that the resulting sequence of sampled values $\{\boldsymbol{\theta}^k\}$ is a Markov chain with stationary distribution $\pi(\boldsymbol{\theta}|\mathbf{y})$.

The idea behind the Gibbs sampler algorithm is that joint probability densities can be completely characterized by their component conditional densities. So, rather than sample from the target density, we will sample from the lower-dimensional *full conditional* densities that together characterize the joint density.

Consider partitioning the parameter vector $\boldsymbol{\theta}$ into d blocks or sub-vectors, $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_d)'$. Then the Gibbs sampler works as follows, with k indexing iterations:

1. **for** $k = 1$ `to` K **do**
2. `sample` $\boldsymbol{\theta}_1^{(k+1)}$ `from` $g_1(\boldsymbol{\theta}_1 \,|\, \boldsymbol{\theta}_2^{(k)}, \boldsymbol{\theta}_3^{(k)}, ..., \boldsymbol{\theta}_d^{(k)}, \mathbf{y})$.
3. `sample` $\boldsymbol{\theta}_2^{(k+1)}$ `from` $g_2(\boldsymbol{\theta}_2 \,|\, \boldsymbol{\theta}_1^{(k+1)}, \boldsymbol{\theta}_3^{(k)}, ..., \boldsymbol{\theta}_d^{(k)}, \mathbf{y})$.
4. `...`
5. `sample` $\boldsymbol{\theta}_d^{(k+1)}$ `from` $g_d(\boldsymbol{\theta}_d \,|\, \boldsymbol{\theta}_1^{(k+1)}, \boldsymbol{\theta}_2^{(k+1)}, ..., \boldsymbol{\theta}_{d-1}^{(k+1)}, \mathbf{y})$.
6. $\boldsymbol{\theta}^{k+1} \leftarrow (\boldsymbol{\theta}_1^{(k+1)}, \boldsymbol{\theta}_2^{(k+1)}, ..., \boldsymbol{\theta}_d^{(k+1)})'$
7. **end for**

2.6.4 Combined use of both algorithms

Gibbs sampling can be considered a variant of the Metropolis-Hastings algorithm in the sense that each component of $\boldsymbol{\theta}$ is updated sequentially and the implicit proposal distributions are simply the full conditional densities $\pi(\boldsymbol{\theta}_j|\boldsymbol{\theta}_{-j}^{(k-1)}, \mathbf{y})$. With this choice, from (2.17), $r = 1$ and each candidate point is always accepted.

The Metropolis-Hastings algorithm is often used in conjunction with a Gibbs sampler for those components of $\boldsymbol{\theta}$ that have conditional distribution that can be evaluated, but can not be sampled from directly, typically because the distribution is known only up to a scale factor. This is exactly our study case, in fact the Bayes theorem for continuous parameters says that

$$\pi(\boldsymbol{\theta}|\mathbf{y}) = \frac{f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}} \propto f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}), \tag{2.18}$$

the constant of proportionality is $\Omega = \left[\int f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}\right]^{-1} = [m_Y(\mathbf{y})]^{-1}$, that is the reciprocal of the marginal distribution of $\mathbf{y}$.

Therefore, even if we don't know the exact density (we know it up to a scale factor Ω) we can use anyway the Metropolis-Hastings algorithm because in the computation of the acceptance rate r we do not need to know Ω since it will be simplified. From (2.17),

$$r = \frac{\pi(\boldsymbol{\theta}^*|\mathbf{y})\, Q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(k-1)})}{\pi(\boldsymbol{\theta}^{(k-1)}|\mathbf{y})\, Q(\boldsymbol{\theta}^{(k-1)}|\boldsymbol{\theta}^*)} = \frac{\cancel{\Omega} f(\mathbf{y}|\boldsymbol{\theta}^*)\pi(\boldsymbol{\theta}^*)\, Q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(k-1)})}{\cancel{\Omega} f(\mathbf{y}|\boldsymbol{\theta}^{(k-1)})\pi(\boldsymbol{\theta}^{(k-1)})\, Q(\boldsymbol{\theta}^{(k-1)}|\boldsymbol{\theta}^*)}$$

All that is required is that we have some approximating density Q from which it is possible to sample, and then it will be able to evaluate the ratio r with the sampled candidate point.

In the following sections we will make a large use of this technique, we will use Gibbs sampler algorithms with Metropolis-Hastings steps in order to sample from the full conditional distributions that are not analytically obtainable.

2.7 Posterior distribution for model A

The original intent was to simulate from the posterior density using the Gibbs sampler algorithm, that is basically sampling the parameters one at time from their full conditional distribution and then update the values. Since we do not have all the explicit forms of the conditional posterior distributions, we are using some Metropolis-Hastings steps within the Gibbs sampler to simulate from the unknown densities.

2.7.1 Conditional posterior distributions

Jump rate λ

$$\begin{aligned}\pi(\lambda|\mathbf{y}, J, s_1, \mathbf{m}) &\propto f(\mathbf{y}|J, \lambda, s_1, \mathbf{m})\pi(\lambda|J, s_1, \mathbf{m}) \\ &\propto f(\mathbf{y}|J, \lambda, s_1, \mathbf{m})\pi(\mathbf{m}|J, \lambda, s_1)\pi(\lambda|J, s_1) \\ &\propto \pi(\mathbf{m}|\lambda)\pi(\lambda)\end{aligned}$$

We can get an explicit form for this density by making some computations:

$$\begin{aligned}\pi(\mathbf{m}|\lambda)\pi(\lambda) &= \left\{\prod_{i=1}^{n} \text{Poi}(\lambda g_i)\right\} \text{Gamma}(\alpha, \beta) \\ &\propto \left\{\prod_{i=1}^{n} \frac{(\lambda g_i)^{m_i} e^{-\lambda g_i}}{m_i!}\right\} \lambda^{\alpha} e^{-\beta\lambda} \\ &\propto \lambda^{(\alpha+\sum m_i)} e^{-(\beta+\sum g_i)\lambda}\end{aligned}$$

that is the kernel of a gamma distribution, thus

$$\lambda|\mathbf{y}, J, s_1, \mathbf{m} \sim \text{Gamma}(\alpha + n\bar{m}, \beta + n\bar{g})$$

where $\mathbf{y} = (y_1, ..., y_n)^T$, $\mathbf{m} = (m_1, ..., m_n)^T$ with mean $\bar{m}$ and finally, $\bar{g} = \frac{1}{n}\sum_{i=1}^{n} g_i$.

Initial state s_1

$$\begin{aligned}P(s_1 = 1|\mathbf{y}, J, \lambda, \mathbf{m}) &= \frac{f(\mathbf{y}|J, \lambda, s_1 = 1, \mathbf{m})P(s_1 = 1)}{f(\mathbf{y}|J, \lambda, s_1 = 1, \mathbf{m})P(s_1 = 1) + f(\mathbf{y}|J, \lambda, s_1 = 0, \mathbf{m})P(s_1 = 0)} \\ &= \frac{f(\mathbf{y}|J, \lambda, s_1 = 1, \mathbf{m})\, p}{f(\mathbf{y}|J, \lambda, s_1 = 1, \mathbf{m})\, p + f(\mathbf{y}|J, \lambda, s_1 = 0, \mathbf{m})\,(1 - p)} \\ &= \hat{p}\end{aligned}$$

Sampling

It is easy to sample from the full conditionals of λ and s_1, because they are known distributions (a Gamma and a Bernoulli), the only computational task is to update the parameters but we can easily obtain independent samples.

Number of jumps in the i'th interval m_i

$$\begin{aligned}
\pi(m_i|\mathbf{y}, J, \lambda, s_1, \mathbf{m}_{-i}) &\propto f(\mathbf{y}|J, \lambda, s_1, \mathbf{m})\pi(m_i|J, \lambda, \mathbf{m}_{-i}, s_1) \\
&\propto f(\mathbf{g}_{\cdot 1}|J, \lambda, s_1, \mathbf{m})\pi(m_i|\lambda) \\
&\propto \prod_{j=1}^{n} f(g_{j1}|J, \lambda, s_j, m_j)\pi(m_i|\lambda) \\
&\propto f(g_{i1}|J, \lambda, s_i, m_i)\pi(m_i|\lambda) \prod_{j=i+1}^{n} f(g_{j1}|J, \lambda, s_j, m_j)
\end{aligned}$$

where $\mathbf{g}_{\cdot 1} = g_{11}, ..., g_{n1}$.

We took away the first i terms of the productory because within them there's no dependency on m_i . We can not do the same with the rest of the terms because every s_j with $j > i$ depends on m_i , remember the relationship $s_k = mod(s_{k-1} + m_{k-1}, 2)$.

Sampling

For the m_i-s we need to use a Metropolis step. Given a current value m_i, we generate a candidate from a shifted Poisson (to force the value to be ≥ 1 not as usual ≥ 0) with mean $m_i - 0.5$. Therefore the proposal density is

$$P(\tilde{m}_i|m_i) = \frac{e^{-(m_i-0.5)}\,(m_i - 0.5)^{(\tilde{m}_i-1)}}{(\tilde{m}_i - 1)!}\ , \quad \tilde{m}_i \geq 1 \tag{2.19}$$

Then we compute the new set of initial states $\tilde{s}_{i+1} = s_i + \tilde{m}_i, \tilde{s}_{i+2} = \tilde{s}_{i+1} + M_i, ...\tilde{s}_n = \tilde{s}_{n-1} + m_{n-1}$. Now we can accept the proposed candidate with probability equal to

$$AR_m = \min\left\{1,\ \frac{f(g_{i1}|\lambda, s_i, \tilde{m}_i)\pi(\tilde{m}_i|\lambda)}{f(g_{i1}|\lambda, s_i, m_i)\pi(m_i|\lambda)} \prod_{j=i+1}^{n} \frac{f(g_{j1}|\lambda, \tilde{s}_j, m_j)}{f(g_{j1}|\lambda, s_j, m_j)} \frac{P(m_i|\tilde{m}_i)}{P(\tilde{m}_i|m_i)}\right\} \tag{2.20}$$

Jump size J

$$\begin{aligned}
\pi(J|\mathbf{y}, \lambda, s_1, \mathbf{m}) &\propto f(\mathbf{y}|J, \lambda, s_1, \mathbf{m})\pi(J|\lambda, s_1, \mathbf{m}) \\
&\propto \frac{1}{J^n}\pi(J) \prod_{i=1}^{n} f(g_{i1} = y_i/J\,|\lambda, J, s_i, m_i)
\end{aligned}$$

Note that the time spent in state 1 in the i'th interval must be less than the total interval length, so that $g_{i1} < g_i$ for $i = 1, ..., n$ hence

$$\frac{y_i}{J} < g_i \Rightarrow J > \frac{y_i}{g_i} \qquad \forall i = 1, ..., n \tag{2.21}$$

defining $J_0 = max_{i=1,...,n}\{\frac{y_i}{g_i}\}$ we have that $J > J_0$.

Sampling

For the jump size we are assuming a gamma prior, $J \sim \text{Gamma}(\eta, \tau)$. Then, we can use a Metropolis step to generate a candidate value. Generate $\log(\tilde{J} - J_0) \sim \text{Normal}\left(\log(J - J_0), \sigma^2\right)$ where σ^2 can be adjusted to achieve an acceptable acceptance rate. Now we accept the proposed candidate with probability proportional to

$$AR_J = \min\left\{1, \frac{\pi(\tilde{J})}{\pi(J)}\frac{J^n}{\tilde{J}^n}\prod_{i=1}^{n}\frac{f(\tilde{g}_{ij}|\lambda, s_i, m_i)}{f(g_{ij}|\lambda, s_i, m_i)}\frac{(\tilde{J} - J_0)}{(J - J_0)}\right\} \tag{2.22}$$

where $\tilde{g}_{i1} = y_i/\tilde{J}$ and $g_{i1} = y_i/J$ for $i = 1, ..., n$.

2.8 The algorithm - model A

We can summarize the algorithm here as follows

- `Fix the function` G.
- `Compute the transformed times` $g_1 = G(t_1), g_2 = G(t_2) - G(t_1), ..., g_n = G(t_n) - G(t_{n-1})$.
- `Compute` $J_0 = \max_i\{y_i/g_i\}$
- `Set initial values` $J^{(0)} > J_0$, $\lambda^{(0)}$, $s_1^{(0)}$, $\mathbf{m}^{(0)}$.
- `For` $i = 1, ..., n$ `compute` $g_{i1}^{(0)} = y_i/J^{(0)}$.
- $h = 0$. `Repeat until convergence is reached:`
  - `Generate` $\lambda^{(h+1)} \sim f\left(\lambda|\mathbf{y}, J^{(h)}, s_1^{(h)}, \mathbf{m}^{(h)}\right)$.
  - `Generate` $s_1^{(h+1)} \sim P\left(s_1 = 1|\mathbf{y}, J^{(h)}, \lambda^{(h+1)}, \mathbf{m}^{(h)}\right)$.
  - `For` $i = 1, ..., n$, `generate` $m_i^{(h+1)}$ `from` $P(m_i|\mathbf{y}, J^{(h)}, \lambda^{(h+1)}, s_1^{(h+1)}, m_1^{(h+1)}, ..., m_{i-1}^{(h+1)}, m_{i+1}^{(h)}, ..., m_n^{(h)})$.

– `Generate` $J^{(h+1)}$ `from` $f(J|\mathbf{y}, \lambda^{(h+1)}, s_1^{(h+1)}, \mathbf{m}^{(h+1)})$.

– `For` $i = 1, ..., n$ `compute` $g_{i1}^{(h+1)} = y_i / J^{(h+1)}$.

– $h = h + 1$.

2.9 Posterior distribution for model B

Even in this case it is too difficult to compute directly the posterior distribution, using the Bayes theorem. Therefore, we are using the same tool to get a random sample from the posterior distribution, a Gibbs sampler algorithm with Metropolis Hastings steps.

2.9.1 Conditional posterior distributions

Initial state s_1

$$\begin{aligned} P(s_1 = 1|\mathbf{y}, J, \mathbf{G}, \mathbf{m}) &= \frac{f(\mathbf{y}|J, \mathbf{G}, s_1 = 1, \mathbf{m}) P(s_1 = 1)}{f(\mathbf{y}|J, \mathbf{G}, s_1 = 1, \mathbf{m}) P(s_1 = 1) + f(\mathbf{y}|J, \mathbf{G}, s_1 = 0, \mathbf{m}) P(s_1 = 0)} \\ &= \frac{f(\mathbf{y}|J, \mathbf{G}, s_1 = 1, \mathbf{m})\, p}{f(\mathbf{y}|J, \mathbf{G}, s_1 = 1, \mathbf{m})\, p + f(\mathbf{y}|J, \mathbf{G}, s_1 = 0, \mathbf{m})\, (1 - p)} \\ &= \hat{p} \end{aligned}$$

Sampling

As already mentioned, it is straightforward to simulate a random sample of s_1 from its full conditional distribution.

Number of jumps in the i'th interval - m_i

$$\begin{aligned} \pi(m_i|\mathbf{y}, J, \mathbf{G}, s_1, \mathbf{m}_{-i}) &\propto f(\mathbf{y}|J, \mathbf{G}, s_1, \mathbf{m}) \pi(m_i|J, \mathbf{G}, \mathbf{m}_{-i}, s_1) \\ &\propto f(\mathbf{g}_{\cdot 1}|J, \mathbf{G}, s_1, \mathbf{m}) \pi(m_i|\mathbf{G}) \\ &\propto \prod_{j=1}^{n} f(g_{j1}|J, \mathbf{G}, s_j, m_j) \pi(m_i|\mathbf{G}) \\ &\propto f(g_{i1}|J, \mathbf{G}, s_i, m_i) \pi(m_i|\mathbf{G}) \prod_{j=i+1}^{n} f(g_{j1}|J, \mathbf{G}, s_j, m_j) \end{aligned}$$

where $\mathbf{g}_{\cdot 1} = g_{11}, ..., g_{n1}$.

We took away the first i terms of the productory because within them there's no dependency on m_i . We can not do the same with the rest of the terms because every s_j with $j > i$ depends on m_i , remember the relationship $s_k = mod(s_{k-1} + m_{k-1}, 2)$.

Sampling

The simulation of the m_i-s is exatcly the same as in the previous model, that is a Metropolis Hastings method to simulate from the full conditional obtained before. The proposal distribution for new $m_i{}^{(k)}$ is a shifted Poisson (to force the value to be ≥ 1 not as usual ≥ 0) with mean $m_i - 0.5$. Then the proposal density is

$$P(\tilde{m}_i|m_i) = \frac{e^{-(m_i-0.5)}\,(m_i-0.5)^{(\tilde{m}_i-1)}}{(\tilde{m}_i-1)!}\ , \quad \tilde{m}_i \geq 1 \tag{2.23}$$

We accept every m_i candidate with probability equal to

$$AR_m = \min\left\{1,\ \frac{P(m_i|\tilde{m}_i)}{P(\tilde{m}_i|m_i)}\frac{f(g_{i1}|\mathbf{G}, s_i, \tilde{m}_i)P(\tilde{m}_i|\mathbf{G})}{f(g_{i1}|\mathbf{G}, s_i, m_i)P(m_i|\mathbf{G})}\prod_{j=i+1}^{n}\frac{f(g_{j1}|\mathbf{G}, \tilde{s}_j, m_j)}{f(g_{j1}|\mathbf{G}, s_j, m_j)}\right\} \tag{2.24}$$

where $\tilde{s}_j$ are the new initial states, since proposing a new candidate value for the number of jumps in the i-th interval could change the following initial states: $\tilde{s}_{i+1} = mod(\tilde{s}_i + \tilde{m}_i, 2)$, $\tilde{s}_{i+2} = mod(\tilde{s}_{i+1} + m_{i+1}, 2)$, ..., $\tilde{s}_n = mod(\tilde{s}_{n-1} + \tilde{m}_{n-1}, 2)$.

Time scale function parameters G

$$\begin{aligned}
\pi(\mathbf{G}|\mathbf{y}, J, \mathbf{m}, s_1) &\propto f(\mathbf{y}|J, \mathbf{G}, s_1, \mathbf{m})\pi(\mathbf{G}|J, \mathbf{m}, s_1) \\
&\propto f(\mathbf{y}|J, \mathbf{G}, s_1, \mathbf{m})\pi(\mathbf{m}|J, \mathbf{G}, s_1)\pi(\mathbf{G}) \\
&\propto f(\mathbf{g}_{\cdot 1}|J, \mathbf{G}, s_1, \mathbf{m})\pi(\mathbf{m}|\mathbf{G})\pi(\mathbf{G}) \\
&\propto \pi(\mathbf{G})\prod_{i=1}^{n} f(g_{i1}|J, \mathbf{G}, s_i, m_i)\pi(m_i|\mathbf{G})
\end{aligned}$$

Note that $m_i|\mathbf{G} \sim \text{Po}(g_i)$, where g_i is $G(t_i) - G(t_{i-1})$.
If we focus on the single parameter, its full conditional is straightforward

$$\begin{aligned}\pi(\mathbf{G}_i|\mathbf{G}_{-i},\mathbf{y},J,\mathbf{m},s_1) &\propto f(\mathbf{y}|\mathbf{G},J,\mathbf{m},s_1)\pi(\mathbf{G}_i|\mathbf{G}_{-i},J,\mathbf{m},s_1)\\ &\propto f(\mathbf{g}_{\cdot 1}|\mathbf{G},J,\mathbf{m},s_1)\pi(\mathbf{m}|\mathbf{G},J,s_1)\pi(\mathbf{G}_i|\mathbf{G}_{-i},J,s_1)\\ &\propto \pi(\mathbf{G}_i)\prod_{j=1}^{n} f(g_{j1}|\mathbf{G},J,m_j,s_j)\pi(m_j|\mathbf{G})\end{aligned}$$

with $\mathbf{G}_i$ we mean any of the four parameters that make up the function, namely (a,b,d,t^*), while $\mathbf{G}_{-i}$ is the set of the three remaining parameters.

Sampling

For what concerns the parameters of the function (a,b,d,t^*) we decided to simulate them one at time with four consequential steps, the reason of this choice is because by simulating the whole set of parameters some problems of acceptance could arise, because it is more complicated to work in a four dimensional space. Therefore, we simulate the parameters from their full conditional densities one at time and at every step we update the $G(t)$ function with the eventually accepted parameted.

Since every parameter has to be strictly positive, we chose a proposal distribution that is a lognormal with mean equal to the previous sampled value and an adjustable standard deviation in order to achieve a good acceptance rate.
Given a current value $\mathbf{G}_i = \gamma$, we generate a candidate $\tilde{G}_i \sim log\,\mathcal{N}(log\,\gamma, \sigma^2)$, thus the proposal density is

$$P(\tilde{G}_i|\mathbf{G}_i = \gamma) = \frac{1}{\tilde{G}_i\,\sigma\sqrt{2\pi}}\, e^{-\frac{(log\,\tilde{G}_i - log\,\gamma)^2}{2\sigma^2}}, \quad \tilde{G}_i > 0 \tag{2.25}$$

After choosing the proposal distribution we have to compute the acceptance rate for the parameters, but there is always the same constraint shown in the previous model that has to be satisfied. Thus, right after generating a new candidate we have to check if it satisfies the condition. If we put in evidence the g_i value (the only that depends on the parameters), we obtain the right constraint:

$$\tilde{g}_i > \frac{y_i}{J}, \quad \forall i \tag{2.26}$$

where $\tilde{g}_i$ is $\tilde{G}(t_i) - \tilde{G}(t_{i-1})$
Then, in conclusion, we have two acceptance/rejection steps for these parameters, one related to the constraint imposed by the model and the second

due to the Metropolis step (here below we have its acceptance rate).

$$AR_{G_i} = \min\left\{1,\ \frac{P(\mathbf{G}_i|\tilde{G}_i)}{P(\tilde{G}_i|\mathbf{G}_i)}\frac{\pi(\tilde{\mathbf{G}}_i)}{\pi(\mathbf{G}_i)}\prod_{j=1}^{n}\frac{f(g_{j1}|\tilde{\mathbf{G}},J,m_j,s_j)\pi(m_j|\tilde{\mathbf{G}})}{f(g_{j1}|\mathbf{G},J,m_j,s_j)\pi(m_j|\mathbf{G})}\right\} \quad (2.27)$$

where $\mathbf{G}$ is the current set of parameters, while $\tilde{\mathbf{G}}$ is the updated set of parameters (in practice we are changing only one of the four G_i at time).

Jump size J

$$\begin{aligned}\pi(J|\mathbf{y},\mathbf{G},s_1,\mathbf{m}) &\propto f(\mathbf{y}|J,\mathbf{G},s_1,\mathbf{m})\pi(J|\mathbf{G},s_1,\mathbf{m})\\ &\propto \frac{1}{J^n}\pi(J)\prod_{i=1}^{n} f(g_{i1}=y_i/J\,|\mathbf{G},J,s_i,m_i)\end{aligned}$$

Sampling

The last parameter to be sampled is the jump size J. We follow the previous algorithm procedure to sample from its full conditional distribution.

Given the current value J we generate $\tilde{J}$ from a lognormal distribution, imposing the constraint, shown in (2.21), that $\tilde{J} > J_0$.
In other words we generate $\log(\tilde{J}-J_0) \sim \text{Normal}\,(\log(J-J_0),\sigma^2)$ where σ^2 can be adjusted to achieve an acceptable acceptance rate. Therefore, the proposal density function is

$$P(\tilde{J}|J) = \frac{1}{(\tilde{J}-J_0)\,\sigma\sqrt{2\pi}}\,e^{-\frac{(log\,\tilde{J}-log\,J)^2}{2\sigma^2}}, \quad \tilde{J} > J_0. \quad (2.28)$$

In conclusion, we accept the candidate $\tilde{J}$ with a probability equal to

$$AR_J = \min\left\{1,\ \frac{\pi(\tilde{J})}{\pi(J)}\frac{J^n}{\tilde{J}^n}\frac{(\tilde{J}-J_0)}{(J-J_0)}\prod_{i=1}^{n}\frac{f(\tilde{g}_{i1}|G,s_i,m_i)}{f(g_{i1}|G,s_i,m_i)}\right\} \quad (2.29)$$

where $\tilde{g}_{i1} = y_i/\tilde{J}$ and $g_{i1} = y_i/J$ for $i = 1,...,n$.

2.10 The algorithm - model B

- `Set initial values for the time scale function` $G^{(0)}$: $a^{(0)}, b^{(0)}, d^{(0)}, t^{*\,(0)}$.
- `Compute` $c^{(0)} = a^{(0)} + (b^{(0)} - d^{(0)})t^{*\,(0)}$.

- Compute the transofmed times $g_1^{(0)} = G^{(0)}(t_1)$,
  $g_2^{(0)} = G^{(0)}(t_2) - G^{(0)}(t_1)$, ..., $g_n^{(0)} = G^{(0)}(t_n) - G^{(0)}(t_{n-1})$.
- Compute $J_0^{(0)} = max_i\{y_i/g_i^{(0)}\}$.
- Set initial values for the other parameters: $J^{(0)} > J_0^{(0)}, s_1^{(0)}, \mathbf{m}^{(0)}$.
- For $i = 1, ..., n$ compute $g_{i1}^{(0)} = y_i/J_0$.
- $h = 0$. Repeat until convergence is reached:
  - Generate $s_1^{(h+1)} \sim P\left(s_1 = 1|\mathbf{y}, J^{(h)}, \mathbf{m}^{(h)}, \mathbf{G}^{(h)}\right)$.
  - For $i = 1, ..., n$, generate $m_i^{(h+1)}$ from
    $P(m_i|\mathbf{y}, J^{(h)}, s_1^{(h+1)}, m_1^{(h+1)}, ..., m_{i-1}^{(h+1)}, m_{i+1}^{(h)}, ..., m_n^{(h)}, \mathbf{G}^{(h)})$.
  - Generate $\mathbf{G}^{(h+1)}$ by updating its parameters one at time:
    - Generate $a^{(h+1)}$ from $\pi\left(a|\,\mathbf{y}, J^{(h)}, s_1^{(h+1)}, \mathbf{m}^{(h+1)}, b^{(h)}, d^{(h)}, t^{*\,(h)}\right)$.
    - Generate $b^{(h+1)}$ from $\pi\left(b|\,\mathbf{y}, J^{(h)}, s_1^{(h+1)}, \mathbf{m}^{(h+1)}, a^{(h+1)}, d^{(h)}, t^{*\,(h)}\right)$.
    - Generate $d^{(h+1)}$ from $\pi\left(d|\,\mathbf{y}, J^{(h)}, s_1^{(h+1)}, \mathbf{m}^{(h+1)}, a^{(h+1)}, b^{(h+1)}, t^{*\,(h)}\right)$.
    - Generate $t^{*\,(h+1)}$ from $\pi\left(t^*|\,\mathbf{y}, J^{(h)}, s_1^{(h+1)}, \mathbf{m}^{(h+1)}, a^{(h+1)}, b^{(h+1)}, d^{(h+1)}\right)$.
    - Compute $c^{(h+1)} = a^{(h+1)} + (b^{(h+1)} - d^{(h+1)})t^{*\,(h+1)}$.
    - Set $\mathbf{G}^{(h+1)} = (a^{(h+1)}, b^{(h+1)}, d^{(h+1)}, t^{*\,(h+1)})$.
  - For $i = 1, ..., n$, update the transofmed times $g_1^{(h+1)} = G^{(h+1)}(t_1)$,
    $g_2^{(h+1)} = G^{(h+1)}(t_2) - G^{(h+1)}(t_1)$, ..., $g_n^{(h+1)} = G^{(h+1)}(t_n) - G^{(h+1)}(t_{n-1})$.
  - Generate $J^{(h+1)}$ from $\pi\left(J|\mathbf{y}, s_1^{(h+1)}, \mathbf{m}^{(h+1)}, \mathbf{G}^{(h+1)}\right)$.
  - For $i = 1, ..., n$, compute $g_{i1}^{(h+1)} = y_i/J^{(h+1)}$.
  - $h = h + 1$.

Chapter 3

Approximate Bayesian Computation (ABC)

In this chapter we will have a look at the Bayesian inference from a different point of view, when the likelihood function is not available, introducing the so called approximate Bayesian computation (ABC) following Marjoram et al. (2003) and Marin et al. (2011).

3.1 Introduction

Many stochastic simulation approaches for generating observations from a posterior distribution depend on the knowledge of the likelihood function. In complex probability models the likelihood function is either difficult to derive analytically or computationally prohibitive to evaluate. However, in some of these cases, it may be straightforward to simulate data from the likelihood itself. ABC methods, also known as likelihood-free techniques, provide a solution to those complex problems in which arise calculation problems to obtain the likelihood.

Under the Bayesian point of view, we imagine data $\mathbf{y}$ generated from a model $\mathcal{M}$ determined by parameters $\boldsymbol{\theta}$, the prior density of which is denoted by $\pi(\boldsymbol{\theta})$.

The posterior distribution of interest is $\pi(\boldsymbol{\theta}|\mathbf{y})$, which is given by

$$\pi(\boldsymbol{\theta}|\mathbf{y}) = f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})/f(\mathbf{y}),$$

where $f(\mathbf{y}) = \int f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}$ is the normalizing constant.

In most of the contexts it is impossible to get such posterior distributions, and stochastic simulations are widely applied to generate observations from

the target distribution. Perhaps the simplest apporach for this is the *rejecton method*:

1. Generate $\boldsymbol{\theta}$ from $\pi(\cdot)$.
2. Accept $\boldsymbol{\theta}$ with probability $h = f(\mathbf{y}|\boldsymbol{\theta})$; repeat until the required number of samples is reached.

Following this scheme, we can get observations that follow the posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{y})$, for details see Ripley (1982).

3.2 The ABC method

Rubin (1984) produced in his paper a description of the first ABC algorithm. The original ABC is in fact a special case of a rejection method where the parameter $\boldsymbol{\theta}$ is generated from the prior $\pi(\boldsymbol{\theta})$ and the acceptance is conditional on the corresponding simulation of a sample being 'almost' identical to the (true) observed sample. We can summarize this first approach in the following way:

1. Generate $\boldsymbol{\theta}'$ from the prior distribution $\pi(\cdot)$.
2. Generate $\mathbf{z}$ from the likelihood $f(\cdot|\boldsymbol{\theta}')$.
3. **if** $\mathbf{z} = \mathbf{y}$
4. set $\boldsymbol{\theta}_i = \boldsymbol{\theta}'$.
5. Repeat until the required number of samples is reached.

The outcome resulting from this algorithm $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_N)$ is an iid sample from the posterior distribution since

$$\begin{aligned} f(\boldsymbol{\theta}_i) &\propto \sum_{\mathbf{z}\in\mathcal{D}} \pi(\boldsymbol{\theta}_i) f(\mathbf{z}|\boldsymbol{\theta}_i) \mathbb{1}_{\mathbf{y}}(\mathbf{z}) = \pi(\boldsymbol{\theta}_i) f(\mathbf{y}|\boldsymbol{\theta}_i) \\ &\propto \pi(\boldsymbol{\theta}_i|\mathbf{y}), \end{aligned}$$

where we are assuming that $\mathbf{y}$ take values in $\mathcal{D}$ (finite or countable set).

Pritchard et al. (1999) extend the above algorithm to the case of continuous sample spaces, producing the first genuine ABC algorithm, defined as follows

1. Generate $\boldsymbol{\theta}'$ from the prior distribution $\pi(\cdot)$.
2. Generate $\mathbf{z}$ from the likelihood $f(\cdot|\boldsymbol{\theta}')$.

3. **If** $\rho\{\eta(\mathbf{z}), \eta(\mathbf{y})\} \leq \varepsilon$

4. `set` $\boldsymbol{\theta}_i = \boldsymbol{\theta}'$.

5. `Repeat until the required number of samples is reached.`

where the parameters of the algorithm are

- η, a function on $\mathcal{D}$ defining a statistic which most often is not sufficient,
- $\rho > 0$, a distance on $\eta(\mathcal{D})$,
- $\varepsilon > 0$, a tolerance level.

The likelihood-free algorithm above thus samples from the marginal in $\mathbf{z}$ of the joint distribution

$$\pi_\varepsilon(\boldsymbol{\theta}, \mathbf{z}|\mathbf{y}) = \frac{\pi(\boldsymbol{\theta}) f(\mathbf{z}|\boldsymbol{\theta}) \mathbb{1}_{A_{\varepsilon,\mathbf{y}}}(\mathbf{z})}{\int_{A_{\varepsilon,\mathbf{y}} \times \Theta} \pi(\boldsymbol{\theta}) f(\mathbf{z}|\boldsymbol{\theta}) \, d\mathbf{z} \, d\boldsymbol{\theta}}, \tag{3.1}$$

where $\mathbb{1}_B(\cdot)$ denotes the indicator function of the set B and

$$A_{\varepsilon,\mathbf{y}} = \{\mathbf{z} \in \mathcal{D} \,|\, \rho\{\eta(\mathbf{z}), \eta(\mathbf{y})\} \leq \varepsilon\}.$$

The basic idea behind ABC is that using a representative (enough) summary statistic η coupled with a small (enough) tolerance ε should produce a good (enough) approximation to the posterior distribution, namely that

$$\pi_\varepsilon(\boldsymbol{\theta}|\mathbf{y}) = \int \pi_\varepsilon(\boldsymbol{\theta}, \mathbf{z}|\mathbf{y}) \, d\mathbf{z} \approx \pi(\boldsymbol{\theta}|\mathbf{y}).$$

3.3 MCMC-ABC

The succes of this approach depends on the fact that the underlying stochastic model $\mathcal{M}$, that is the conditional density $f(\cdot|\boldsymbol{\theta})$, is easy to simulate. In addition it is important that data at the proposal stage are located in high posterior probability regions; however this is very unlikely when we are simulating from a non informative prior distribution $\pi(\cdot)$. To make this algorithm more efficient, Marjoram et al. (2003) introduce an MCMC-ABC algorithm targeting the approximate posterior distribution π_ε of (3.1).

1. `Use the standard ABC method to get a realization` $(\boldsymbol{\theta}^{(0)}, \mathbf{z}^{(0)})$ `from the target distribution` $\pi_\varepsilon(\boldsymbol{\theta}, \mathbf{z}|\mathbf{y})$.

2. **for** $k = 1$ `to` N **do**

3. `Generate` $\boldsymbol{\theta}'$ `from the Markov kernel` $q(\cdot|\boldsymbol{\theta}^{(k-1)})$,

4. `Generate` $\mathbf{z}'$ `from the likelihood` $f(\cdot|\boldsymbol{\theta}')$,

5. `Generate` u `from` $\mathcal{U}(0,1)$,

6. **if** $u \leq \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}^{(k-1)}|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(k-1)})q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(k-1)})}$ `and` $\rho\{\eta(\mathbf{z}'), \eta(\mathbf{y})\} \leq \varepsilon$

7. `set` $(\boldsymbol{\theta}^{(k)}, \mathbf{z}^{(k)}) = (\boldsymbol{\theta}', \mathbf{z}')$

8. **else**

9. `set` $(\boldsymbol{\theta}^{(k)}, \mathbf{z}^{(k)}) = (\boldsymbol{\theta}^{(k-1)}, \mathbf{z}^{(k-1)})$,

10. **end for**

The acceptance probability used in this algorithm does not involve the calculation of the likelihood and thus satisfies ABC requirements. It also produces an MCMC algorithm which exactly targets $\pi_\varepsilon(\boldsymbol{\theta}, \mathbf{z}|\mathbf{y})$ as its stationary distribution; for the complete proof, see Marjoram et al. (2003).

3.4 Calibration of ABC

As noted before, the ABC approximation depends on tuning parameters (the summary statistic η, the tolerance level ε and the distance ρ) that have to be chosen prior to running the algorithm.

The tolerance ε is somewhat the easiest aspect of this calibration issue in that, when ε goes to zero, the ABC algorithm becomes exact and gives us samples from the real posterior distribution,

$$\pi_\varepsilon(\boldsymbol{\theta}, \mathbf{z}|\mathbf{y}) \to \pi(\boldsymbol{\theta}|\mathbf{y}) \quad \text{when } \varepsilon \to 0.$$

As noted above, the choice of the tolerance level ε is mostly a matter of computational power: smaller ε's are associated with higher computational costs and the standard practice is to select ε as a small percentile of the simulated distances $\rho\{\eta(\mathbf{z}), \eta(\mathbf{y})\}$.

Several authors have considered the fundamental difficulty associated with the choice of the summary statistic $\eta(\mathbf{y})$, which one would like to consider as a quasi-sufficient statistic. Unfortunately for most real problems it is impossible to find sufficient statistics. Furthermore, the summary statistics

of interest are usually determined by the problem at hand and chosen by the experimenters in the field.

Many simulation experiments were done comparing ABC-MCMC and the original ABC algorithm, for details see McKinley et al. (2009). The authors have also tested strategies to select the tolerance level, and to choose the distance ρ and the summary statistics. The conclusions are not very surprising, in that

1. repeating simulations of the data points given one simulated parameter does not seem to contribute to an improved approximation of the posterior by the ABC sample,

2. the tolerance level does not seem to have a strong influence,

3. the choice of the distance, of the summary statistics and of the calibration factors are paramount to the success of the approximation, and

4. ABC-MCMC outperforms ABC.

3.5 ABC and model choice

Model choice is one particular aspect of Bayesian analysis that involves computational complexity, if only because several models are considered simultaneously. In addition to the parameters of each model, the inference considers the model index $\mathcal{M}$, which is associated with its own prior distribution $\pi(\mathcal{M} = m),\ (m = 1, ..., M)$ as well as a prior distribution on the parameters conditional on the value m of the model index, $\pi_m(\boldsymbol{\theta}_m)$, defined on the parameter space Θ_m. The choice between these models is then driven by the posterior distribution of $\mathcal{M}$, a challenging computational target where ABC brings a straightforward solution. Indeed, once $\mathcal{M}$ is incorporated within the parameters, the ABC approximation to the posterior follows from the same principles as regular ABC, as shown by the following pseudo-code, where $\boldsymbol{\eta}(\mathbf{z}) = (\eta_1(\mathbf{z}), ..., \eta_M(\mathbf{z}))$ is the concatenation of the summary statistics used for all models (with elimination of duplicates).

1. **for** $i = 1$ `to` N **do**
2. **repeat**
3. `Generate` m `from the prior` $\pi(\mathcal{M} = m)$

4. Generate $\boldsymbol{\theta}_m$ from the prior $\pi_m(\boldsymbol{\theta}_m)$

5. Generate $\mathbf{z}$ from the model $f_m(\mathbf{z}|\boldsymbol{\theta}_m)$

6. **until** $\rho\{\eta(\mathbf{z}), \eta(\mathbf{y})\} < \varepsilon$

7. Set $m^{(i)} = m$ and $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}_m$

8. **end for**

The ABC estimate of the posterior probability $\pi(\mathcal{M} = m|\mathbf{y})$ is then the acceptance frequency from model m, namely

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{m^{(i)}=m} \ .$$

This also corresponds to the proportion of simulated datasets that are closer to the data $\mathbf{y}$ than the tolerance ε.

3.6 Applications

In this section we apply ABC techniques to our specific medical problem. We will use both MCMC-ABC and ABC algorithms.

3.6.1 ABC for model A

To set up an ABC algorithm we have to choose the prior distribution for the parameters and the tuning parameters ρ, η, ε. We are using the same prior distribution as in the Gibbs sampler for model A, thus

$$\pi(J, \lambda, s_1, \mathbf{m}) = \pi(J)\,\pi(s_1)\,\pi(\lambda, \mathbf{m}) = \pi(J)\,\pi(s_1)\,\pi(\mathbf{m}|\lambda)\,\pi(\lambda)$$

- $J \sim \text{Gamma}(\eta, \tau)$
- $\lambda \sim \text{Gamma}(\alpha, \beta)$
- $s_1 \sim \text{Be}(p)$
- $\pi(\mathbf{m}|\lambda) = \prod_{i=1}^{n} \pi(m_i|\lambda) = \prod_{i=1}^{n} \text{Poi}(\lambda g_i)$

where $g_i = G(t_i) - G(t_{i-1})$.

The other choice that we have to do before running the ABC is that of the summary statistic, the distance and the tolerance level. We decided to

fix the tolerance a posteriori, in other words we first run the simulation for N values and then we compute the distance and keep the values that have a distance from the real data less than the 5th percentile. Namely $\varepsilon = \rho_{0.05}$, where $\rho_{0.05}$ is the empirical 5th percentile. The distance ρ is the sum of square distances between the values, therefore the summary statistic is just the value itself.

To summarize we fixed:

- $\varepsilon = \rho_{0.05}$,
- $\eta(\mathbf{y}) = \mathbf{y}$,
- $\rho(\eta(\mathbf{y}), \eta(\mathbf{z})) = \rho(\mathbf{y}, \mathbf{z}) = \sum_{i=1}^{n}(y_i - z_i)^2$.

Then, in this specific case, the algorithm becomes

1. **for** $k = 1$ `to` N **do**
2. `Generate` $J^{(k)}$ `from a Gamma(`η, τ`).`
3. `Generate` $\lambda^{(k)}$ `from a Gamma(`α, β`).`
4. `Generate` $s_1^{(k)}$ `from a Bernoulli(`p`).`
5. `Generate` $m_i^{(k)}$ `from a Poisson(`g_i`), for every` $i = 1, ..., n$.
6. `Generate` $\mathbf{z}^{(k)}$ `from` $f(\cdot|J^{(k)}, \lambda^{(k)}, s_1^{(k)}, \mathbf{m}^{(k)})$.
7. `Compute` $\rho^{(k)} = \sum_{i=1}^{n}(y_i^{(k)} - z_i^{(k)})^2$.
8. **end for**
9. `Set` ε `as the 5th percentile of` $\rho^{(k)}$.
10. `Select only` $\hat{k}$`-s that satisfy:` $\rho^{(\hat{k})} \leq \varepsilon$.
11. `And we get the sample` $J^{(\hat{k})}$ `that is an approximation of a random sample from` $\pi(J|\mathbf{y})$.

This approach could give us values of J very far from the 'real' value because we are sampling from an arbitrary non informative prior, then it may be hard to explore the whole state space unless we choose a very small tolerance level, but in that case will arise computational problems.

3.6.2 MCMC-ABC for model A

For the motivations mentioned before, we decided to improve the approach of the simple ABC by including a Metropolis step.

This approach is very similar, the only difference is that we sample from a proposal density (instead that from the prior) and we add an acceptance step as in the usual Metropolis algorithm, therefore the output will be a Markov chain because there is dependency between the samples.

In this situation we have to fix the proposal density to generate new samples of the parameters, we decided to keep using the same proposal density used for J in the previous algorithms, that is a lognormal density with mean equal to the previous value $J^{(k-1)}$ with the usual constraint that $J > J_0$. In this case we must fix the tolerance ε a priori, and then we decided to use the same as in the ordinary ABC (also we kept the same distance and summary statistic) for sake of simplicity. The scheme of this approach is:

1. Fix the initial values $J^{(0)}, s_1^{(0)}, \mathbf{m}^{(0)}$.
2. **for** $k = 1$ to N **do**
3. Generate $s_1^{(k)}$ from a Bernoulli(p).
4. Generate $m_i^{(k)}$ from a Poisson(g_i), for every $i = 1, ..., n$.
5. Generate $\tilde{J}$ from the proposal density $q(\cdot|J^{(k-1)})$.
6. Generate $\mathbf{z}^{(k)}$ from the lilkelihood $f(\cdot|\tilde{J}, s_1^{(k)}, \mathbf{m}^{(k)})$.
7. Generate u from $\mathcal{U}(0, 1)$.
8. Compute $AR_J = \frac{\pi(\tilde{J})q(J^{(k-1)}|\tilde{J})}{\pi(J^{(k-1)})q(\tilde{J}|J^{(k-1)})}$.
9. **if** $u \leq AR_J$ and $\rho(\mathbf{z}^{(k)}, \mathbf{y}) \leq \varepsilon$
10. set $J^{(k)} = \tilde{J}$
11. **else**
12. set $J^{(k)} = J^{(k-1)}$
13. **end for**

There is a trade off between the precision of the method and computational cost: in fact the standard ABC involves less calculations than the MCMC-ABC method, but it may require more time to run because without the MC part the normal ABC approach will propose values from the prior distribution, in some cases the high density region of the posterior distribution is far away from the one of the prior, thus it will take a lot of time to get an acceptable number of samples sufficiently 'close' to the real data in terms of the distance ρ.

3.6.3 ABC for model B

As we did for model A, to apply approximate bayesian computation we need to fix the prior distribution for the parameters and the tuning parameters. In this case we took the prior already used for traditional bayesian computations (the Gibbs sampler algorithm for model B) and we kept the same ABC tuning parameters that we used with model A.

$$\pi(J, s_1, \mathbf{m}, \mathbf{G}) = \pi(J, s_1)\,\pi(\mathbf{m}, \mathbf{G}) = \pi(J)\,\pi(s_1)\,\pi(\mathbf{m}|\mathbf{G})\,\pi(\mathbf{G})$$

- $J \sim \text{Gamma}(\eta, \tau)$
- $s_1 \sim \text{Be}(p)$
- $\pi(\mathbf{m}|\mathbf{G}) = \prod_{i=1}^{n} \pi(m_i|\mathbf{G}) = \prod_{i=1}^{n} \text{Poi}(g_i)$
- $\pi(\mathbf{G}) = \pi(a)\,\pi(b)\,\pi(d)\,\pi(t^*) = \prod_j \text{Gamma}(\alpha_j, \beta_j)$

With tuning parameters namely

- $\varepsilon = \rho_{0.05}$,
- $\eta(\mathbf{y}) = \mathbf{y}$,
- $\rho(\eta(\mathbf{y}), \eta(\mathbf{z})) = \rho(\mathbf{y}, \mathbf{z}) = \sum_{i=1}^{n}(y_i - z_i)^2$.

Then the algorithm is straightforward,

1. **for** $k = 1$ `to` N **do**
2. `Generate` $J^{(k)}$ `from a Gamma(`η, τ`).`
3. `Generate` $a^{(k)}$ `from a Gamma(`α_1, β_1`).`
4. `Generate` $b^{(k)}$ `from a Gamma(`α_2, β_2`).`

5. Generate $d^{(k)}$ from a Gamma(α_3, β_3).

6. Generate $t^{*(k)}$ from a Gamma(α_4, β_4).

7. Generate $s_1^{(k)}$ from a Bernoulli(p).

8. Generate $m_i^{(k)}$ from a Poisson(g_i), for every $i = 1, ..., n$.

9. Generate $\mathbf{z}^{(k)}$ from $f(\cdot|J^{(k)}, \mathbf{G}^{(k)} = (a^{(k)}, b^{(k)}, d^{(k)}, t^{*(k)}), s_1^{(k)}, \mathbf{m}^{(k)})$.

10. Compute $\rho^{(k)} = \sum_{i=1}^{n}(y_i^{(k)} - z_i^{(k)})^2$.

11. **end for**

12. Set ε as the 5th percentile of $\rho^{(k)}$.

13. Select only $\hat{k}$-s that satisfy: $\rho^{(\hat{k})} \leq \varepsilon$.

14. And we get the sample $(J^{(\hat{k})}, a^{(\hat{k})}, b^{(\hat{k})}, d^{(\hat{k})}, t^{*(\hat{k})})$ that is an approximation of a random sample from $\pi(J, a, b, d, t^*|\mathbf{y})$.

3.6.4 Model selection with ABC

Here we consider a wider range of models, in fact using approximate bayesian computation we can work with all the models of which we do not know the likelihood function but we are able to simulate data from them. In particular we are interested in finding the optimal model in terms of number of states $\{0, 1, ..., k\}$.

We are considering five different models in terms of state space $\mathcal{S}$, every different model has $\mathcal{S} = \{0, ..., k\}$, where $k = 1, ..., 5$. For every model we consider the function $G(t)$ fixed after the LS estimation. In this case we have to put a prior distribution not only on the parameter set $(J, \mathbf{m}, s_1)$, but we shall add the prior distribution of the model index $\pi(\mathcal{M})$. We kept using the same prior as in the previous examples for the parameters. For what concerns the prior of $\mathcal{M}$, since we do not have any preferred model we decided to take as prior a discrete uniform over the set of the five models just mentioned.

We are using the same tuning parameters as in the previous algorithms, namely

- $\varepsilon = \rho_{0.05}$, the tolerance is fixed as the 5th percentile of the distance.
- $\eta(\mathbf{y}) = \mathbf{y}$.
- $\rho(\eta(\mathbf{y}), \eta(\mathbf{z})) = \rho(\mathbf{y}, \mathbf{z}) = \sum_{i=1}^{n}(y_i - z_i)^2$.

The model selection algorithm includes three simulation steps: first we sample the index of the model from which we are going to simulate from, then there is the sampling of the parameters of that specific model (in our case it is straightforward because all the involved models share the same parameters, that are the jump size J, the initial state s_1 and the number of jumps in each interval m_i), after that there is the most important phase, sample data from the selected model with the sampled parameters. After these three simulation steps we have to choose from the whole sampled sets the ones 'closer' to the real data in terms of distance ρ, to do that we are evaluating the distance for each sampled data, ordering the data in terms of increasing distance and taking only the first 5% of the data.

1. **for** $k = 1$ `to` N **do**
2. `Generate the model index` $q^{(k)}$ `from a discrete uniform.`
3. `Generate` $J^{(k)}, s_1^{(k)}, \mathbf{m}^{(k)}$ `from` $\pi(\cdot)$.
4. `Generate` $\mathbf{z}^{(k)}$ `from the` $q^{(k)}$`-th model fixing` $(J, s_1, \mathbf{m}) = (J^{(k)}, s_1^{(k)}, \mathbf{m}^{(k)})$.
5. `Compute` $\rho^{(k)} = \sum_{i=1}^{n} (y_i^{(k)} - z_i^{(k)})^2$.
6. **end for**
7. `Set` ε `as the 5th percentile of` $\rho^{(k)}$.
8. `Select only` $\hat{k}$`-s that satisfy:` $\rho^{(\hat{k})} \leq \varepsilon$.
9. `Now we focus our attention to` $q^{(\hat{k})}$ `that is an approximation of the posterior distribution of the models` $\pi(\mathcal{M} = q|\mathbf{y})$.

The estimate of the posterior probability is then the acceptance frequency of model q.

Chapter 4

Examples and applications

In this chapter we will see some applications of the models introduced up to now. At first we will explain in detail two different procedures to simulate data from their conditional likelihood. After that we will describe the two data sets used throughout all this chapter for inference purposes: one is a real data set, while the other is a simulated one.

Our applications are basically divided into three areas: the first two consists of Bayesian analysis through the standard MCMC algorithms described in Chapter 2, applied to both the simulated and the real data set. In the last part we will use approximate Bayesian computation applied to the simulated data set, to compute Bayesian inference in order to compare results with the traditional MCMC approach.

4.1 Simulating data from the model

In this section we will explain in detail how to simulate growth curve data $\{Y_t\}$. We need to generate growth curve data, from the conditional distribution (2.11), for two reasons: first we can see how the output curve changes when we modify some parameters in (2.11); secondly because we will need simulated data sets to test the algorithms introduced in the previous chapters.

There are two different ways of carrying out the simulation of a curve, the first is more intuitive, while the second is more computationally efficient but less flexible because it is applicable only in the simple model described by (2.1).

We will make a comparison between the different simulated data sets, in order to have a general idea of how changing the parameters in (2.11) affects the generated data.

4.1.1 Simulation I

Here we start from the definition of the process $\{Y_t\}$, introduced in (1.3), in order to get a procedure for simulating this kind of processes.

Remember that $Y_t = V_{G(t)} = J\int_0^{G(t)} U_s\, ds$, where $\{U_s\}$ is a birth-death process with equal birth and death rates λ and state space $\{0, 1, 2, ..., k\}$. Then we know that the time between two jumps is exponentially distributed with parameter λ and, since birth and death rates are the same, the probability to jump forward is the same of jumping backward. Therefore, since we assume that $G(t)$ is known, we can just sample a realization from the $\{U_s\}$ process and then compute the cumulative integral, after fixing J, to get the trajectory of the target process $\{Y_t\}$.

It is straightforward to get a realization of $\{U_s\}$: it is a step function so that we only need to know the initial state, the times when the process jumps from one state to another and the direction of the jump.

We can summarize the procedure as follows:

1. Fix the function $G(t)$, jump rate λ, number of states k and jump size J.
2. Sample the initial state s_1 from a discrete uniform distribution over $\{0, 1, 2, ..., k\}$, that means that $\mathbb{P}(s_1 = i) = 1/k \quad i = 0, 1, 2, ..., k$.
3. Set $J_0 = 0$ and $U_0 = s_1$.
4. Compute the time of the jumps J_i in a window of time $[0, T_{\max}]$.
   - **while** $J_i < T_{\max}$
   - Sample a random time $T_i \sim \text{Exp}(\lambda)$.
   - Define $J_{i+1} = J_i + T_i$.
5. For every jump, draw a random variable to select the direction of the jump:
   - **if** $U_{J_i} = 0$ **then** $D = 1$.
   - **if** $U_{J_i} = k$ **then** $D = -1$.
   - **else** $D = \pm 1$ with probability 1/2.
   - Define $U_{J_{i+1}} = U_{J_i} + D$.
6. U_t is defined for every $t \in [0, T_{\max}]$ as the step process starting with a value equal to s_1 and changing state every J_i as just defined.

7. Compute $Y_t = \int_0^{G(t)} U_s \, ds$.

In real applications we do not have continuous growth curves, since time is always discretized in some way. However, using this the first type of simulation we obtain the whole process, that can be easily evaluated at every time to get an array of values if requested.

4.1.2 Simulation II

The second method to sample from (2.11) concerns only the two state model $\mathcal{S} = \{0, 1\}$, that is model A introduced in the previous chapter. For this model we are able to compute the likelihood function given all the parameters; therefore here we are using that likelihood to get random realizations of Y_t.

We start fixing the time interval $[0, T_{\max}]$; if we define

$$Z_t = \{\#\text{ of jumps at time } t \geq 0\},$$

we have that the process $\{Z_t\}_{t\geq 0}$ is a Poisson process of rate λ, because the holding times of the process are exponentially distributed with parameter λ.

Exploiting the transition probability definition we can obtain the distribution of $m_i|\lambda, G(t)$, where $m_i = Z_{G(t_i)} - Z_{G(t_{i-1})}$ represents the number of jumps inside the (transformed) i-th interval. All the increments are independent and for each t, $m_i \sim \text{Poi}(\,\lambda(G(t_i) - G(t_{i-1}))\,)$. From the second chapter (see (2.3)) we know the likelihood function of $\mathbf{y}$, that is the vector of the increments of heights. Then, if we fix all the parameters we can generate a sample from its conditional distribution. To do that we need to previously get a sample of the m_is, but it is straightforward once we know that they follow a Poisson distribution.

This is the scheme for the second method:

- Fix the function $G(t)$, jump rate λ, and jump size J.
- Choose the time grid $\{t_i\}_{i=1,\dots,n}$ where to evaluate the process $\{Y_t\}$.
- Compute $g_i = G(t_i) - G(t_{i-1})$ for all $i = 1, ..., n$ $(G(t_0) = 0)$.
- Sample the initial state s_1 from a Bernoulli(1/2).
- For $i = 1, ..., n$:
  - sample m_i from a Poisson(λg_i)

- Sample g_{i1} from the conditional likelihood.
- Compute $y_i = Jg_{i1}$

- Then define $Y_{t_i} = \sum_{j \leq i} y_j$ for $i = 1, ..., n$.

4.1.3 Simulated data sets from the conditional density (2.11)

Now that we are able to simulate growth curves from the model we will focus our attention to see how the parameters (jump rate λ, jump size J) and the state space $\mathcal{S}$ influence the output curve. We are using *simulation I*.

In order to better understand the role of every single parameter we decided to fix all of them (and the time scale function) but one and see how the output curves change with every single parameter. In the following plots we will see what happens when the jump size, space state and jump rate are changing while the other parameters remain fixed.

We show realizations of from the conditional likelihood with different parameter values. Figure 4.1(d) shows several replications of the real children growth data, while the other three plots (a)-(b)-(c) show several realizations of the same process, changing one parameter at time. In all of these simulations the time scale function $G(t)$ was estimated by least square error method applied to the real data, and fixed.

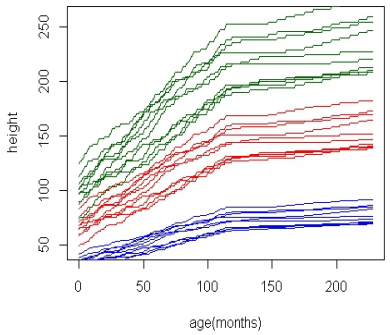
We can see that when the jump size is lower the mean value of the curve is lower (remember that J is the multiplier of the integral in the definition given by (2.1). In addition the realizations show smoother trajectories with less variability between them when the jump size is lower.

The situation is very similar if we fix J and let the state space vary, in fact increasing the number of states we get more variability between curves and a higher mean value.

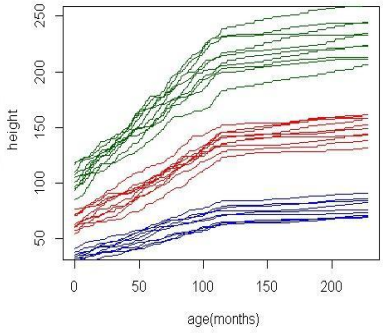
The jump rate does not show any evident impact on the curves. We will see later that this parameter can be omitted from the model.

4.2 Non-decreasing longitudinal data sets

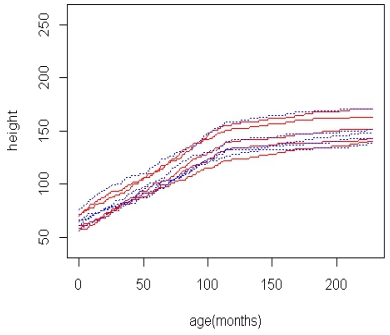
By modeling growth curves we can work with different kinds of longitudinal data with the only constraint that the curves must be non-decreasing. This approach is useful in clinical studies (e.g. monitoring height, weight or other increasing variables) but also in other fields such as reliability problems or stress tests (think for instance to a degradation process of a machinery).



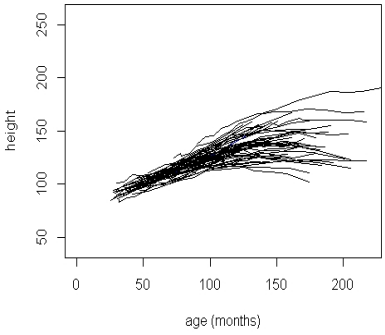
(a) $\lambda = 1$, $\mathcal{S} = \{0, 1\}$ and J varying ($J = 1$ in blue, $J = 2$ red, $J = 3$ green)



(b) $\lambda = 1$, $J = 1$ and $\mathcal{S} = \{0, ..., k\}$ where $k = 1$ (blue), $k = 2$ (red), $k = 3$ (green)



(c) $J = 2$, $\mathcal{S} = \{0, 1\}$ and $\lambda = 1$ (blue), $\lambda = 25$ (red)



(d) Real growth curves

Figure 4.1: Simulations of Y_t with different parameters, compared to real data.

During the following examples we will apply all the methods and algorithms introduced in the previous chapters. To compare the results and test the performances of our different ways of modeling we will use two data sets: the first is a real data set taken from clinical data, while the second is a simulated data set (created using *simulation II*).

4.2.1 Simulated data set: `data1`

This data set is a simulated one: we used *simulation II* to obtain a collection of fifty curves made by twenty observations.

This data set will be useful to test the validity of our models, because to simulate it we fixed all the non latent parameters: $J, \lambda, G(t)$. Therefore, we will have the possibility to see if our outputs (mainly posterior distributions) are acceptable according to the 'real' parameters.

In the following table there are the values that we fixed to generate the data set. The choice of the G parameters was made after a least square estimation over the real data set `data2`).

Table 4.1: Parameters fixed for the simulation of `data1`

parameter	value
J	2.50
λ	10.00
a	63.33
b	0.64
c	124.10
d	0.11
t^*	114.61

4.2.2 Real data set: `data2`

The other data set is taken from the analysis of the data collected on children suffering from acute lymphoblastic leukemia (ALL). ALL is a form of leukemia, or cancer of the white blood cells characterized by excess lymphoblasts. Malignant, immature white blood cells continuously multiply and are overproduced in the bone marrow. ALL causes damage and death by crowding out normal cells in the bone marrow, and by spreading (infiltrating) to other organs. ALL is most common in childhood with a peak incidence at 2–5 years of age, and another peak in old age.

All children affected from ALL need chemotherapy into the cerebrospinal fluid (CSF) to kill any leukemia cells that might have spread to the brain and spinal cord. This treatment, known as *intrathecal chemotherapy*, is given through a lumbar puncture (spinal tap). It is usually given twice (more often if the leukemia is high risk) during the first month and 4 to 6 times during the next 1 or 2 months. It is then repeated less often during the rest of consolidation and maintenance.

In one of the clinical trials carried out at Dana Farber Cancer Institute (Boston, USA), a total of 618 children were treated between November 1987 and December 1995 with three different central nervous system therapies: intrathecal therapy alone (no radiation), intrathecal therapy with conventional cranial radiation, and intrathecal therapy with twice daily radiation. Measurements on height were taken at diagnosis and approximately every 6 months thereafter. Previous studies on the effects of cranial radiation on height suggested that radiation contributed to decreased expected height, since cranial radiation has been associated with the development of growth hormone deficiency.

The purpose of this analysis is to evaluate the long-term effects of treatment on the children height and on the individual growth trajectories. In Figure 4.2 are represented the growth curve of the patients, divided into groups. Patient in the second and third group are treated with cranial radiations, one aim of the study is to statistically confirm or not the relationship between cranial radiation and decreased expected height that has been suggested in previous medical studies.

In Figure 4.2 we can see the data divided into groups, the numerosity of the groups is not the same. Even the length of the curves is not the same for every patient. Another important fact that we have to manage is the presence of some measurement errors, in fact there are some decreasing curves that we are going to delete from the original data set to build the one used for doing analysis.

To pass from clinical data to our data set we had to make a selection from the curves, some of them are affected by measurement errors (for instance there are some decreasing curves, that is impossible since we are facing with children growth curves). Thus, we first 'cleaned' our data set in order to delete curves affected from an evident error. After that we selected thirty curves from each group, we took the curves with the highest number of observations to better interpolate values from them. At the end of this step for every curve we interpolated twenty equally spaced points over a time grid.

In conclusion we have our real data set (from now on we are referring to it as `data2`), composed of three sets of thirty curves, each made by twenty equally spaced observations (the same time grid was used for `data1`).

The main difference between `data1` and `data2` is that the first consists in a single group of curves, thus it will be used to test the goodness of the models introduced, while the second data (the real one) is divided into three groups, it will be used to study the variability between groups.

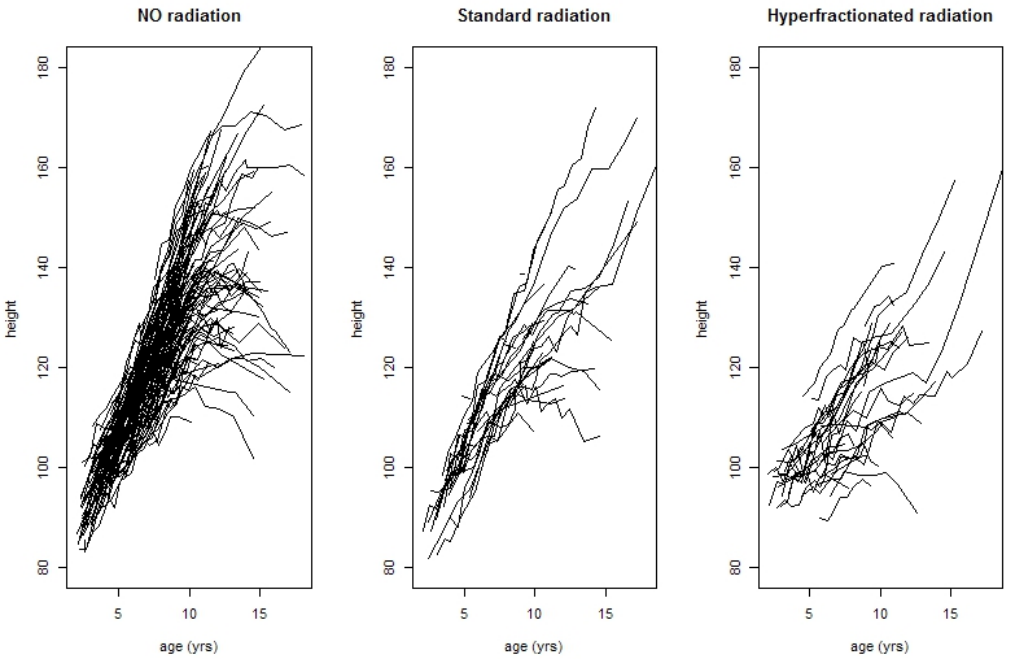


Figure 4.2: Height of the children over time for each treatment received.

4.3 Example 1: Bayesian inference for `data1`

In this example we will consider `data1`, our simulated data set with fixed parameters. We will estimate posterior densities of the parameters, after that we will be able to make a comparison between the Bayesian estimates of the quantities of interest and the 'real' values fixed for the simulation.

4.3.1 Posterior estimates: model A

From the previous simulations, the jump rate λ seems to be not so relevant for the estimation of the process: we might be facing with a non identifiable parameter. Some preliminary trials show that the Markov chain of λ is not converging, then we computed the moment generating function of the process and look for the contribution of the jump rate parameter in the moments of the growth process. The result is that the jump rate does not influence any of the moments. Therefore we cannot obtain information about it from the data.

Moments of the process

Here we compute the moments of the process $\{Y_t\}$ in the case of a two states model (0 and 1) with equal jump rate λ for both directions of jump. Following the general formula obtained in (1.9) we can get the moment generating function of this special case.

In the case of the two states model the Markov process has a stationary distribution with same probabilities $\Pi_0 = \Pi_1 = \frac{1}{2}$ because the jump rate is the same. Thus the Laplace transform is

$$f_Y^*(s) = \frac{1}{2} + \frac{1}{2}\, e^{-sJG(t)} \tag{4.1}$$

The jump rate parameter is not present in the Laplace transform, so it also will not contribute in any of the n-th derivatives, therefore λ does not take part in any of the n-th moments of the process, so it is impossible to estimate it and its real value is useless for other inference purposes.

From now on we will always fix the jump rate parameter, it will be irrelevant for us. We chose to use a non-informative prior for J, setting the parameters of the Gamma as (1/2,1/2). For what concerns λ, we fixed it to 1 (then the prior on λ becomes a Dirac function centered on 1).

We ran the algorithm for model A for 60,000 iterations with a burn-in period of 10,000 iterations, for a final sample size of 50,000. The results are summarized in Figure 4.3.

We can see that the chain has converged, the autocorrelation between consequent samples is low enough to consider the output as independent realizations from the posterior density of J.

The credible interval for J (with probability 95%) is $(2.341, 2.686)$. This interval includes the real value of J that is 2.5.

4.3.2 Posterior estimates: model B

Here we will lead to an analysis similar to the one done before. The main difference is that now we are not estimating a time scale function a priori, but we will try to obtain independent realizations from its conditional posterior distribution on the data.

We will run the algorithm described for model B, with our simulated data set.

To test the validity of model B we will apply the Bayesian algorithm and run the Gibbs sampler for a total of 60,000 iterations, with an initial burn-in of 10,000 iterations, for a final sample size of 50,000.

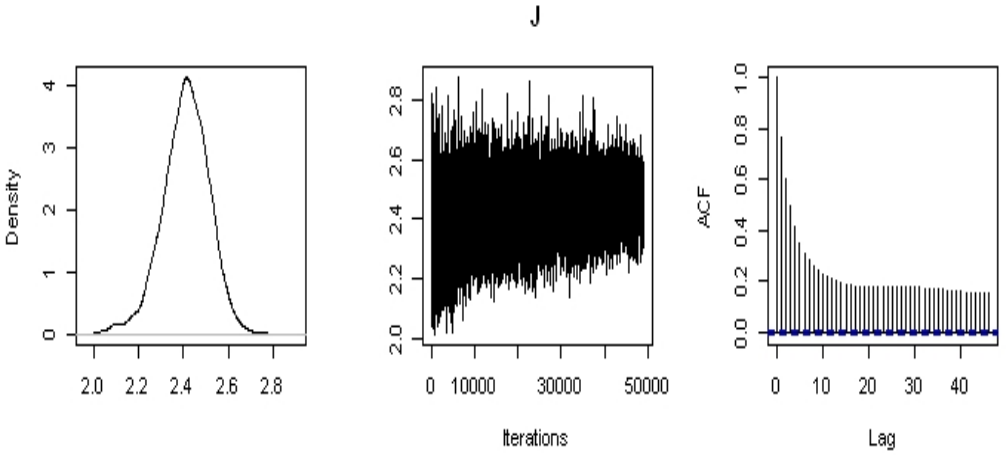


Figure 4.3: Posterior density (left panel), traceplot (central panel) and autocorrelation function (right panel) of the jump size J.

We used a non-informative prior, as for model A. To do that we just fixed all the Gamma parameters to 1/2. Therefore we have that $J, a, b, d, t^* \sim$ Gamma$(1/2, 1/2)$, all independent.

In this model we have five non latent variables as our output from the simulation, the jump size J and the four parameters of $G(t)$. To make a good inference we have to keep in mind that not all these parameters are independent, some of them are coupled and so we will not take the parameters as they are but we will control some combination of them, to make inference for some important quantities of interest such as the mean value of the curve.

Rewriting the formula for the mean value of the process in function of the parameters of the model we obtain

$$E[Y_t] = \frac{J}{2}G(t) = \begin{cases} \frac{Ja}{2} + \frac{Jb}{2}t, & \text{for } t \leq t^* \\ \frac{Jc}{2} + \frac{Jd}{2}t, & \text{for } t > t^*, \end{cases} \tag{4.2}$$

therefore, since we will make inference on the mean value of the curve we will rearrange our output parameters to get a sample composed by $\left\{\left(\frac{Ja}{2}\right)^{(k)}, \left(\frac{Jb}{2}\right)^{(k)}, \left(\frac{Jd}{2}\right)^{(k)}, t^{*(k)}\right\}$. Then, as we did before, we can obtain c by the other four parameters (a, b, d, t^*) and compute also $\left(\frac{Jc}{2}\right)^{(k)}$.

As we can see, in many cases the estimates are very different from the values fixed. The reason is because in this case there is an overparametrization, in fact if we change our scope and reparametrize our model in function

Table 4.2: In this table we compare the values of the parameters, on the left side there are the 'real' values fixed to generate the data set while on the right side there are their estimates. We took as estimate the MAP for each parameter (mean a posteriori).

parameter	value	estimate
a	63.33	13.71
b	0.64	0.12
c	124.1	25.95
d	0.108	0.02
t^*	114.61	114.26
J	2.5	10.02

of the mean value we can obtain a new set of parameters and use them to estimate the mean process $E[Y_t]$.

Table 4.3: In this table we compare the values of the new parameters, obtained by simply recombine the exsisting ones in function of the mean process.

parameter	real value	mean	st.dev.	2.5%	50%	97.5%
$(Ja)/2$	79.16	71.50	14.34	57.86	70.06	88.51
$(Jb)/2$	0.80	0.60	0.180	0.50	0.58	1.41
$(Jc)/2$	155.12	128.75	18.65	109.21	126.38	161.35
$(Jd)/2$	0.13	0.10	0.03	0.08	0.09	0.19
t^*	114.61	114.26	3.64	110.54	114.29	119.70

Now we can work with this new set of parameters and forget about the previous one, without loss of information. In Table 4.3 we collected the information about this new set of parameters, the ones that make up the mean trajectory of the process. We can see that every credible interval contains the real value and that there is a higher influence of the standard deviation for the parameters related to the intercept rather than the slope.

In Figure 4.4 we compared the two mean trajectories, in black the real one and in red the estimate one. To build the estimate growth curve we took the posterior mean value for every parameter and then compute analitically the curve, following (4.2). The estimate growth curve is underestimating the real one, but we can notice that this underestimation is related to a bad estimate of the intercept values. In fact the slopes seem to be very similar;

even the position of the t^* point (where there is a change in the growth speed) is estimated well.

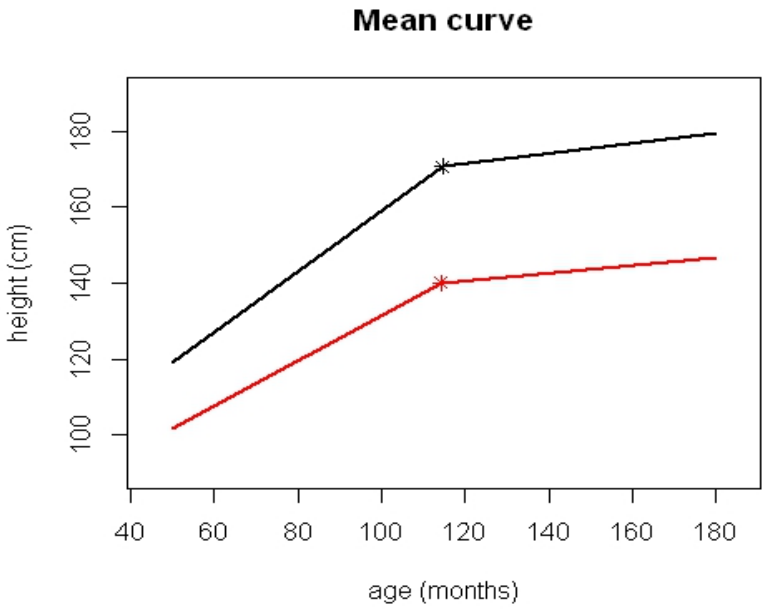


Figure 4.4: Mean growth curve of the growth process (black) and its estimate (red).

4.4 Example 2: Bayesian inference for `data2`

In this section we will make inference with the real data set, modeled with models A.2 and B.2. Both models that we will test are for grouped data set, model A.2 includes three different jump sizes and a fixed time scale function; model B.2 expects only one jump size, the same for each group, but three different time scale functions.

Our goal is to see possible differences between the groups, in terms of expected growth or growing trend. Medical studies suggest that the second and third group present a slower growth phase, because of the radiation treatment to which the patients were exposed to.

4.4.1 Posterior estimates: model A.2

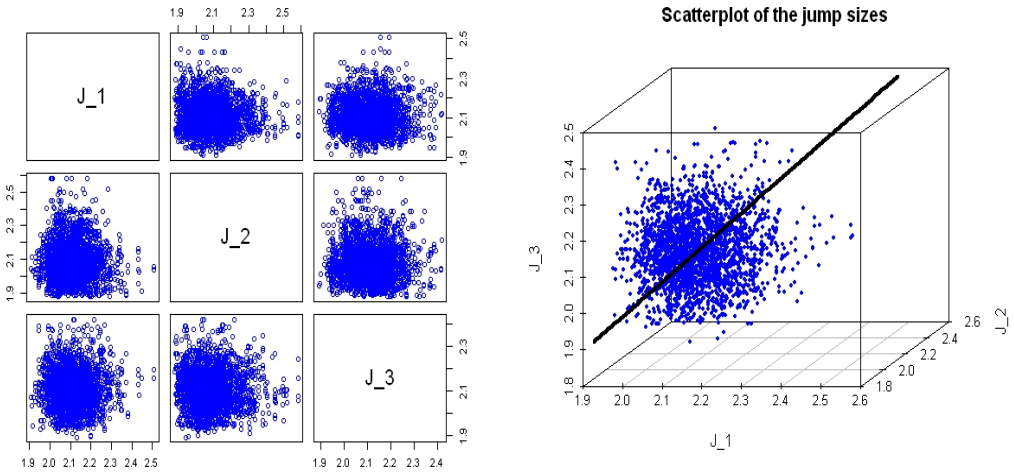
The previous example, with a simulated data set, helped us to better understand the model, then we realized that the jump rate λ can be fixed without loss of generality or equally we can run the algorithms without monitoring that parameter.

Now we are trying to estimate the jump size J from the real data set, that is divided into three groups.

We will use model A.2 in this section, setting the prior as the same for example I but with the introduction of a 3-dimensional vector of jump sizes $\mathbf{J}$, for its prior we chose the product of three Gamma with parameters (1/2, 1/2). Choosing that prior we are assuming independence between the jump sizes of every group keeping the non-informative choice that we kept using in all the examples.

The target distribution now is the 3-dimensional posterior of $\mathbf{J}|\mathbf{Y}$ that we can obtain by marginalizing the posterior of the parameters, resulting from the algorithm.

We ran a total of 110,000 iterations, after a burn-in of 10,000 iterations, so that the total sample size is 100,000.



(a) Scatter matrix

(b) Three dimensional scatter plot. In black is plotted the line $J_1 = J_2 = J_3$

Figure 4.5: Two and three dimensional scatter plots of the realizations from the posterior of (J_1, J_2, J_3) given the data

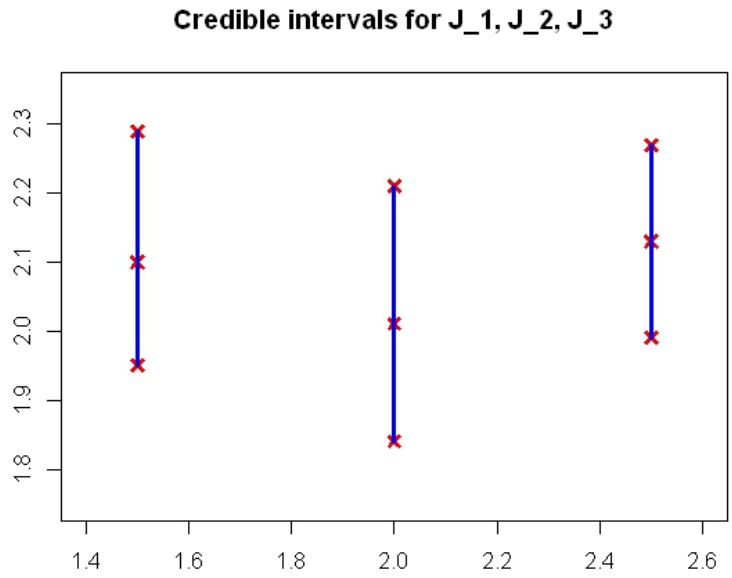


Figure 4.6: 95% posterior credible intervals for J_1, J_2, J_3

From the scatter plots and the credible intervals of the jump sizes (Figures 4.5 and 4.6) we can see that the three groups seem to have different parameters, i.e. the posterior does not concentrate on the $J_1 = J_2 = J_3$ line. In fact, if we compare the credible intervals we can notice that the second group presents a jump size that is lower than the other two. Therefore when we analyze our data we have to distinguish between groups, because the jump size J_i is related to the mean of the process. We can compute the mean of the process (see (1.6)) and obtain

$$E[Y_t] = \frac{J}{2} G(t).$$

Then, if we take the three posterior means for the three groups we can say that the mean curve is changing passing from one group to another. In this specific example we are assuming $G(t)$ fixed, hence we will have different mean curves but all with the same shape. The jump size affects only the position of the growth curve. We will see in the next example that letting the function vary it is possible to get more satisfying conclusions. Once we fix the shape of the mean curve by fixing $G(t)$ we can conclude that the mean function seem to shift between groups.

4.4.2 Posterior estimates: model B.2

Our data set is divided into three groups, therefore we will lead to a group analysis, imposing three different functions, one $G_i(t)$ for every group, and then we will analize the respective posterior distribution of the parameters $(a_i, b_i, d_i, t_i^*),\ i = 1, 2, 3$. In particular, we are interested to estimate the mean growth curve, that is given by $\frac{J}{2}G(t)$.

In this case we are not assuming three different jump sizes but a shared one, the variability between the groups is expressed by choosing three different time scale functions.

Our aim is to see whether or not there is a difference between the three groups in terms of expected growth, and also if there is such difference we are interested to know the shape of the growth curve (different parameters mean a different growth curve). We will run the algorithm for model B.2, using the real data set (data (1)).

The algorithm is the same described in detail in the second chapter. We will run a Gibbs sampler within Metropolis steps to generate from all the full conditional distribution, in this case we have more parameters to simulate because we are assuming three functions (then twelve parameters instead of four) but the procedure is the same, there will be a slow down in computational speed due to the bigger state space that we are exploring. We chose to set the prior parameters as pure non-informative, setting all the Gamma densities with parameters (1/2, 1/2).

We ran the algorithm for 200,000 iterations after a burn-in period of 50,000, the remaining chain was thinned with a thinning interval of length 15, so that the final sample size is 10,000. We needed more iterations because there are more parameters than in the other examples, therefore the algorithm requires more time to reach convergence.

In Figure 4.7 we can see that our output Markov chain presents a good mixing and the autocorrelation spectrum is good enough to make good inference of the J parameter. The credible interval with probability 95% is $(1.78, 2.31)$.

In Figure 4.9 we can compare the posterior marginal densities of the parameters for each group. Looking at the credible intervals of the intercept a we can see that the first group (children treated without radiation) present a higher value than the other two groups. This means that the growth process of the patients of group one is shifted higher than the other two mean curves.

Another very important parameter is t^*, it is a time point where the time scale function G changes its slope. Physically it can be read as the average age in which the children slows its growth. From the credible intervals it is clear that this age comes first in the two groups treated with radiation

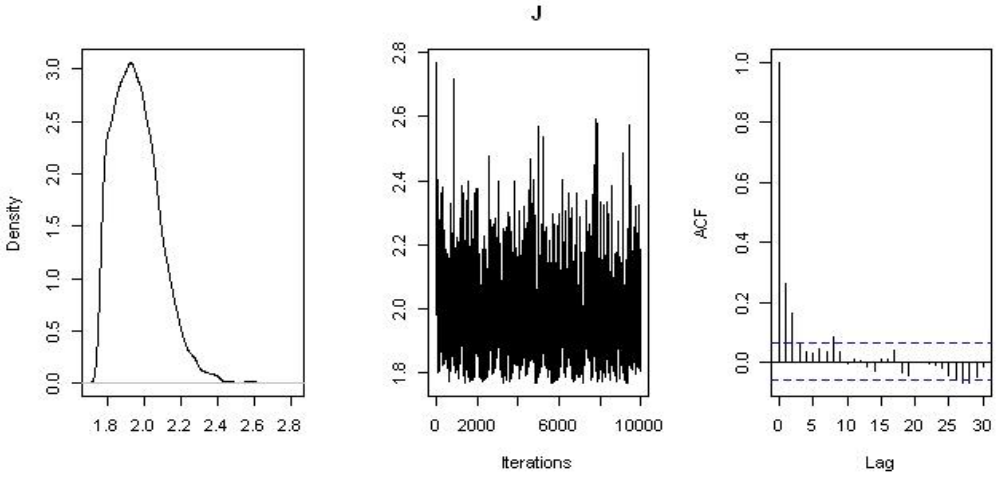


Figure 4.7: Posterior density (left panel), traceplot (central panel) and autocorrelation function (right panel) of the jump size J in the case of grouped data.

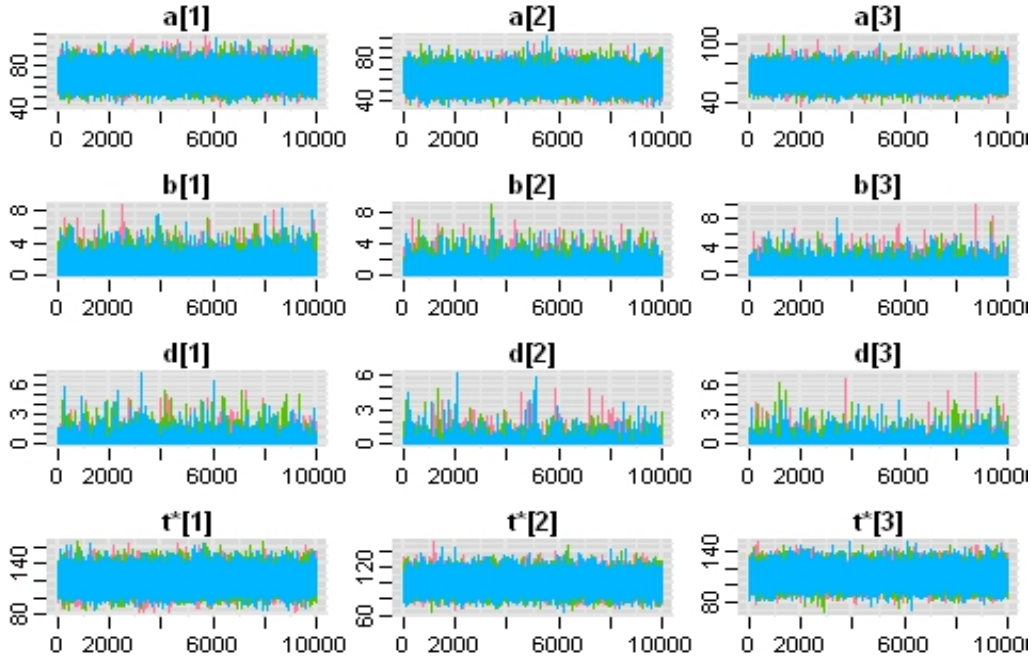
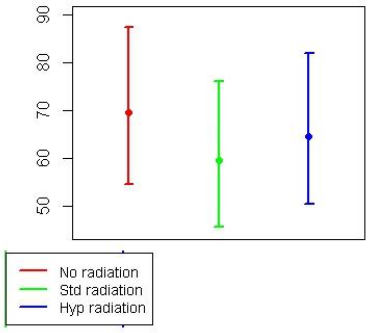
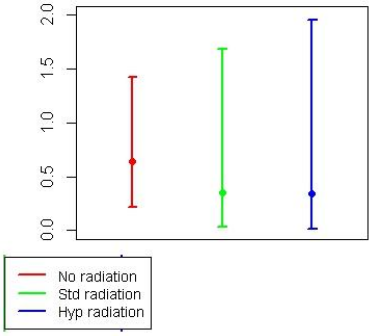


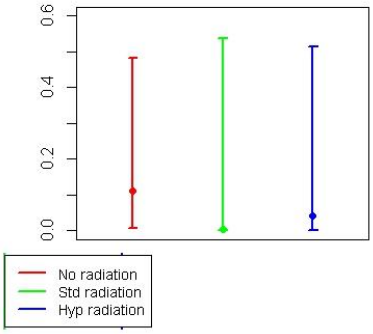
Figure 4.8: Trace plots of the parameters of G, four for every group for a total of twelve parameters.



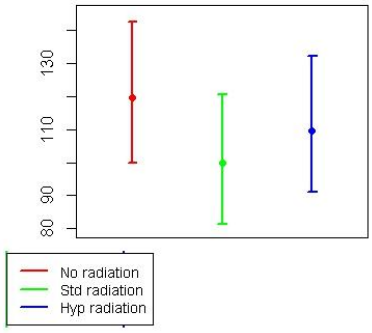
(a) Intercept - a



(b) Slope of the first half of the curve - b



(c) Slope of the second half of the curve - d



(d) Changing slope point - t^*

Figure 4.9: Credible intervals (95%) for the parameters of $G(t)$, compared for the three groups of curves.

(it seems that the second group, standard radiation, has the earliest age of changing growth).

For what concerns b and d, which represent the slope of the curve in the two different growth phases, it is harder to read the output marginal densities. In fact we can notice a higher variability in the last two groups, and also lower values in terms of mean and median. It might be that, as clinical studies already confirm, the growth speed is lower in the patients treated with radiations rather than the other patients.

To have a graphical representation easy to understand, we plotted the

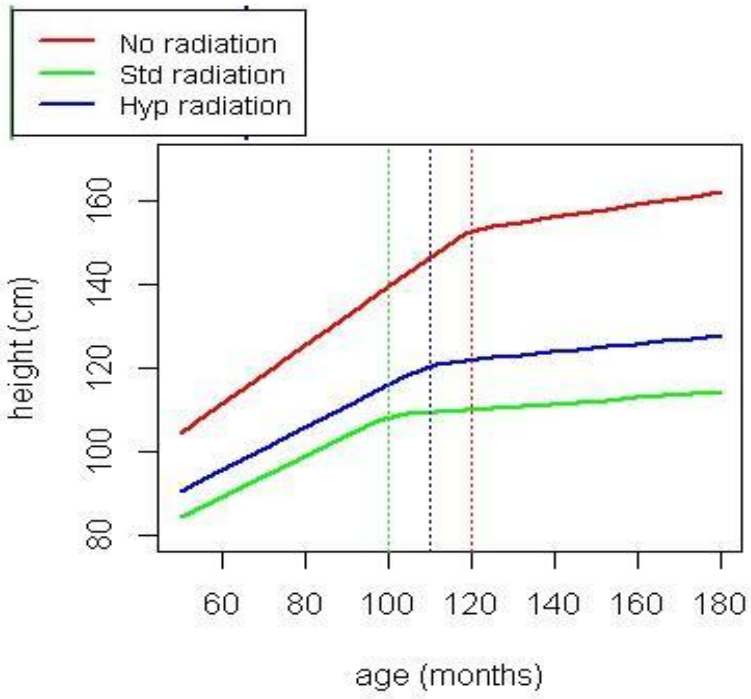


Figure 4.10: Expected growth curves of the three groups compared.

three estimates of the mean growth curves in Figure 4.10. From this plot we can infer that the mean curve of the first group (red curve, representing the group treated without radiotherapy) is widely higher than the other two; considering only the patients treated with radiotherapy we can conclude that the treatment of the third group seems to be less invasive since the expected growth is slowed less.

We can focus our attention also to the position of the t^* point (dashed vertical lines), patients treated with standard radiation start to decrease their growth speed before the others, then it comes the turning point for the 'blue' group (hyperfractioned radiation) and last the t^* for the group treated without radiations.

4.5 Example 3: ABC methods

Here we present the ABC approach to the problem. We will start applying the ABC and MCMC-ABC methods to the data used for example 1 (using `data1`), in particular, we will use ABC with both models A and B and we will use the MCMC-ABC approach with model A. Then we will be able

to compare performances and results of standard Bayesian algorithms and these ones, but we are also interested to see if there will be a significant computational improvement by using the MCMC-ABC instead of the simple ABC.

At the end of all these comparison we will go further and move where the traditional Bayesian approach is not applicable, because we do not have information about the likelihood of such models as in the multi-state case (we have the likelihood in an explicit form only when we are considering the two-state model). Considering different models we will use the ABC for model selection purposes testing the model selection algorithm with the real data set (data (1)).

4.5.1 ABC for model A

In this example we will use the same simulated data set as in example I so we can compare the two outputs, the one resulting from the Gibbs sampler and the one resulting from ABC computation. We will use both ABC and MCMC-ABC approaches.

We are using the algorithm introduced in the previous chapter, fixing the prior distribution in this way:

$$\pi(J, \mathbf{m}, s_1) = \pi(J)\pi(s_1)\prod_{i=1}^{n}\pi(m_i) = \text{Gamma}(1/2, 1/2)\text{Be}(1/2)\prod_{i=1}^{n}\text{Poi}(g_i).$$

We chose those parameters to set a non-informative prior, because we do not have further information about the parameters and we are not sure if they have a physical meaning.

We also ran the algorithm for MCMC-ABC. The prior distribution is the same as before, we have to fix the proposal density to generate new samples of the parameters, we decided to keep using the same proposal density used for J in the previous algorithms, that is a lognormal density with mean equal to the previous value $J^{(k-1)}$ with the usual constraint that $J > J_0$. In this case we must fix the tolerance ε a priori, and then we decided to use the same as in the ordinary ABC (also we kept the same distance and summary statistic) for sake of simplicity.

We ran both algorithms for a total of 200,000 iterations, and selected ε as the 5-th percentile of the simulated distances $\rho\{\eta(\mathbf{z}, \eta(\mathbf{y}))\}$. In this way we have a total sample size of 10,000. for each algorithm.

In Figure 4.11 we can see three estimates of the posterior density of J resulting from the three different approaches: the first one is the Gibbs sampler algorithm while the other two are the ABC and the MCMC-ABC

respectively. The data set was simulated and the 'real' value of the jump size J was fixed to 2.5.

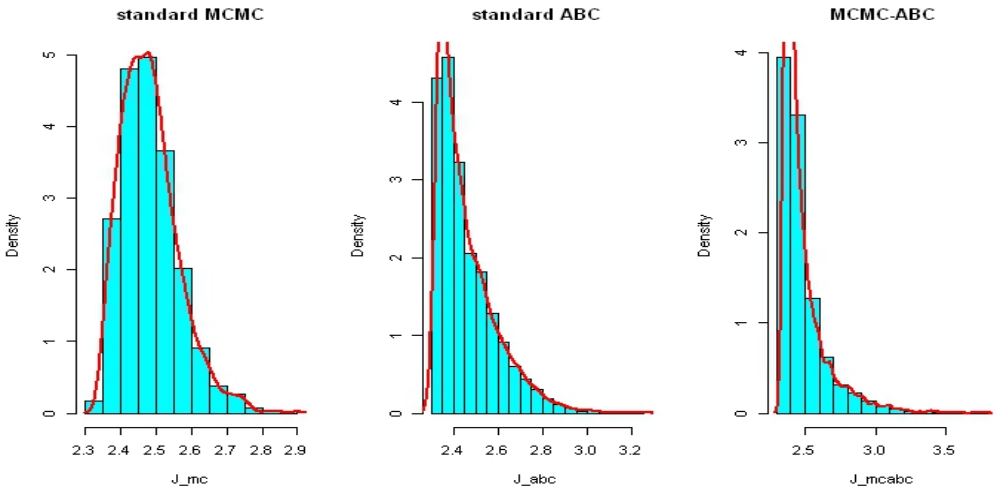


Figure 4.11: Posterior distribution of J, using three different algorithms: MCMC (left panel), ABC (center panel) and MCMC-ABC (right panel).

The first approach is the only one that requires the knowledge of the likelihood function, therefore it is the more accurate but it may be very expensive in terms of computational costs, because at every Metropolis step within the Gibbs sampler it is required to compute the likelihood ratio (to obtain the acceptance rate). The other two ABC methods do not require any likelihood function.

4.5.2 ABC for model B

The same computational approach was applied to model B, in this case we simulated 250,000 samples of (J, a, b, d, t^*) and selected the first 5% of the data in terms of distance. Our final sample of parameters is a collection of 12,500 realizations from the approximate posterior distribution.

The prior distribution of the parameters is:

$$\begin{aligned} \pi(J, \mathbf{G}, \mathbf{M}, \mathbf{s_1}) &= \pi(J)\pi(\mathbf{M}|\mathbf{G})\pi(\mathbf{G})\pi(\mathbf{s_1}) \\ &= \pi(J)\prod_{j=1}^{K}\pi(\mathbf{M}_j|G_j)\pi(G_j)\pi(\mathbf{s_1}_j), \end{aligned}$$

where J, a, b, d, t^* were distributed as four independent Gamma distributions with parameters (1/2, 1/2), $s_1 \sim \text{Be}(1/2)$ and $m_j|G$ were distributed, as usual, as independent Poisson with mean $g_i = G(t_i) - G(t_{i_1})$.

As we did for standard MCMC algorithm with model B, we are interested to estimate the mean curve. Therefore, we recombined the parameters in order to obtain the quantities of interest to compute $E[Y_t]$, namely $((Ja)/2, (Jb)/2, (Jc)/2, (Jd)/2, t^*)$; remember that c is not a variable but just a function of the other parameters of the G curve: $c = a + (b - d)t^*$.

In the following tables we summarize the marginal posterior distribution of these parameters and we also compare these distributions with the results obtained using the Gibbs sampler algorithm for model B.

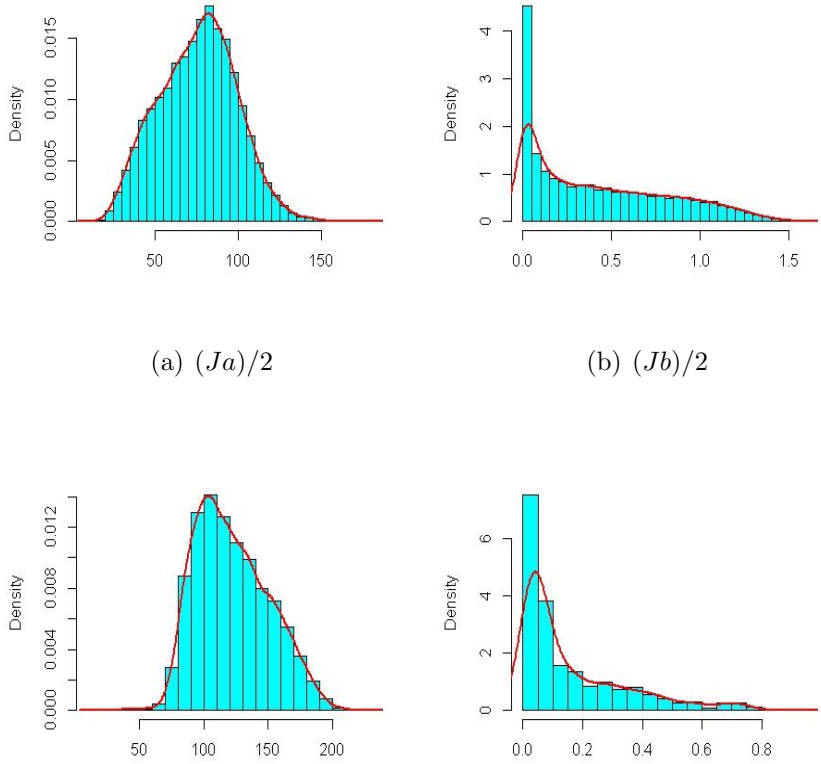
Table 4.4: Comparison between real values and estimate values (posterior means) obtained respectively with MCMC and ABC algorithms.

parameter	real value	mcmc-estimate	abc-estimate
$(Ja)/2$	79.16	71.5	75.83
$(Jb)/2$	0.80	0.60	0.42
$(Jc)/2$	155.12	128.75	122.18
$(Jd)/2$	0.13	0.10	0.09
t^*	114.61	114.26	110.11

Table 4.5: Credible intervals with probability 95% for the recombined parameters of model B.

parameter	2.5%	50%	97.5%
$(Ja)/2$	32.32	77.02	120.15
$(Jb)/2$	0.12	0.34	1.24
$(Jc)/2$	77.89	118.39	181.39
$(Jd)/2$	0.01	0.08	0.67
t^*	108.07	109.99	111.94

We can state that ABC is a suitable method for estimating parameters of this model, it reaches good results compared to the standard MCMC algorithm. In terms of computational cost, ABC requires more iterations to get sufficiently good estimates because all the parameters were sampled from non-informative priors without a following acceptance/rejection step, as in the ordinary Gibbs sampler.



(a) $(Ja)/2$

(b) $(Jb)/2$

(c) $(Jc)/2$

(d) $(Jd)/2$

Figure 4.12: Marginal posterior distributions resulting from ABC.

4.5.3 Model selection with ABC

Here we will see the problem from a different point of view. We will work on the real data set (`data2`) and we wonder whether the two state model is the 'right' one; previously we took always that model for simplicity, because it was the only one with an available likelihood function. We are interested to know if there are other 'good' models with more than two states. To do that we will apply ABC for model selection, working on the real data set. As in every ABC method we will need to simulate data from every model, in the following algorithm we will simulate from the different multi-state models by using *simulation I*, since the other approach to simulate data is not suitable for multi-state models.

We ran a total of 250,000 simulations, to have a final sample of 12,500 observation due to the tolerance level that we set. In Figure 4.13 we can see the posterior probabilities of each model.

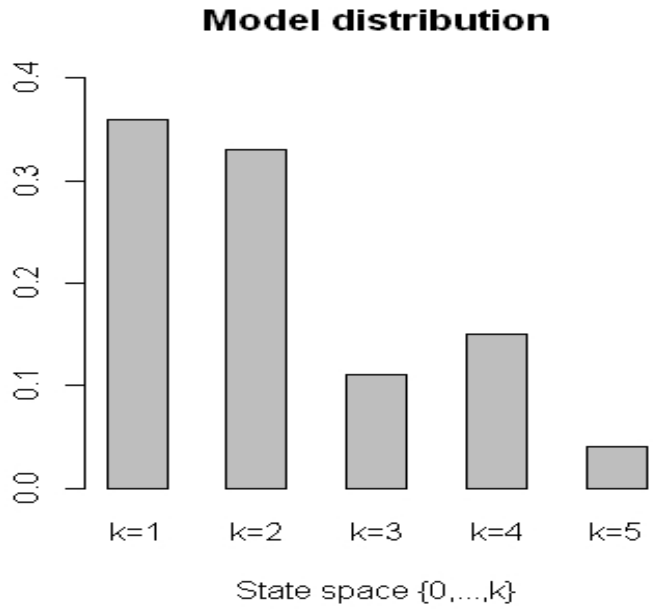


Figure 4.13: Posterior probabilities of the model index; the k-th model has state space $\mathcal{S} = \{0, ..., k\}$, with $k = 1, ..., 5$.

Comparing the posterior probabilities of each model we can see that is more unlikely that data come from a high number of state model; in fact, almost 70% of the total mass of the posterior is concentrated on the first two models: the two and the three state ones.

4.6 Conclusions and further work

In Chapter 4 we have applied our two Bayesian models to a real and a simulated data set (respectively, `data1` and `data2`). In particular, the analysis on `data1` showed robustness of models A and B, while the analysis on the real data was focused on comparing the posterior estimates of the parameters for the three groups of patients.

The analysis of the simulated data led to satisfactory conclusions regarding the validity of the models, especially model A (the one with a fixed

$G(t)$), which has J (the jump size) as its only variable of interest; this variable was estimated correctly in our examples. For what concerns model B, in which we introduced the parameters that characterize $G(t)$, the analysis was a bit more difficult, because when increasing the number of variables, issues related to the convergence of Markov chains could arise. In our specific case, introducing the four parameters of the time-scaling function, there is an identifiability issue, or, from a Bayesian perspective, a multi-modality in the posterior distribution of the parameters; to face this problem we reduced the variability of the transition kernels in the Metropolis-Hastings steps within the algorithms, in order to focus the exploration of the state space around the initial values of the parameters. Given that, it is important that the initial values are chosen in a proper way, thus we decided to estimate the parameters with frequentist techniques and use these estimates as our starting values for the Monte Carlo algorithms.

The analysis on `data2` was carried out imposing the same conditions mentioned above, to avoid other identifiability problems. With real data, divided in three groups, we compared the posterior estimates and we confirmed previous studies that suggested that radiation contributed to decreased expected height, since cranial radiation has been associated with the development of growth hormone deficiency. In addition to having a lower average height, patients treated with radiotherapy enter in a second phase of growth, in which the growth slows, earlier than others. In our models this second phase starts after the time point t^*, where the G function changes its slope.

The statistical performances of our model are strongly affected by the choice of the parametrization of the time-scaling function: in our analyses we chose a specific linear parametrization of the curve, based on a previous study of the data set. Further work could consists in assuming different $G(t)$-s, that could be parametrized or considered as non-parametric elements. By doing the same analysis with a wider range of time-scaling functions (model selection via Bayes Factor or ABC) we could get more reliable results, less related to the original choice of the function.

Further investigations could be the prediction of new observations. We are interested to predict the growth trend for a new patient just entered in the study; in the Bayesian context, this information is usualy obtained via the predictive distribution $\mathcal{L}(Y_{n+1}|Y_1, ..., Y_n)$.

References

Albert, J. (2007). *Bayesian Computation with R*. Springer, New York.

Dalton, V.K., Rue, M., Silverman, L.B., Gelber, R.D., Asselin, B.L., Barr, R.D., Clavell, L.A., Hurwitz, C.A., Moghrabi, A., Samson Y., Schorin, M., Tarbell, N.J., Sallan, S.E., Cohen, L.E. (2003). Height and weight in children treated for acute lymphoblastic leukemia: relationship to CNS treatment. *Journal of Clinical Oncology*, **21**, 2953–2960.

David, H.A., Nagaraja, H.N. (2003). *Order Statistics*. Wiley, New York.

Jackman, S. (2009). *Bayesian Analysis for the Social Sciences*. Wiley, New York.

Durbán, M., Harezlak, J., Wand, P., Carroll, R.J. (2004). Simple fitting of subject-specific curves for longitudinal data. *Statistics in Medicine*, **00**, 1-24.

Kingman, J.F.C. (1993). *Poisson Processes*. Oxford University Press, Oxford.

Marin, J.M., Pudlo, P., Robert, C.P., Ryder, R.J. (2011). Approximate Bayesian computational methods. *Statistics and Computing*, **22**, 1167-1180.

Marjoram, P., Molitor, J., Plagnol, V., Tavaré, S. (2003). Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, **100**(26), 15324-15328.

McKinley, T., Cook, A., Deardon, R. (2009). Inference in epidemic models without likelihoods. *International Journal of Biostatistics*, **5**(1), 24.

Norris, J.R. (1997). *Markov Chains*. Cambridge University Press, Cambridge.

Palacios, A.P. (2012). *Models and inference for population dynamics*. Doctoral thesis, Universidad Carlos III, Madrid.

Pritchard, J., Seielstad, M., Perez-Lezaun, A., Feldman, M. (1999). Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, **16**(12), 1791-1798.

Ripley, B.D. (1982). *Stochastic Simulation*. Wiley, New York.

Robert, C. (2001). *The Bayesian Choice*, Second Edition. Springer, New York.

Rubin, D. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *Annals of Statistics*, **12**(4), 1151-1172.