Politecnico di Milano

Scuola di Ingegneria Indistruale e dell'Informazione

Corso di Laurea Magistrale in Ingegneria Matematica

POLITECNICO
MILANO 1863

A Bayesian autoregressive semiparametric model for waiting times of recurrent events

Relatore: **Prof.ssa Alessandra Guglielmi**
Co-relatore: **Prof.ssa Maria Di Iorio**

Candidato:
Giorgio Paulon
Matr. 816626

Anno Accademico 2014 - 2015

Al papà,
alla mamma

Abstract

In this work we propose a Bayesian approach for the analysis of recurrent event data. The first main original contribution of this thesis consists of a semiparametric Bayesian model for waiting times between recurrent events. In particular, time-dependency of waiting times from the previous ones is modelled through mixtures of autoregressive processes. In addition, this model allows to create clusters of patients according to the entire trajectory of the event counts over the period of observation. Both fixed and time-dependent covariates can be introduced in the present framework. As the second main original contribution of this thesis, we derive the analytical expression of the full-conditional distributions necessary to build an MCMC algorithm to sample from the posterior distribution. The algorithm was efficiently coded in the Julia language. Data in this context usually consist of a large number of processes exhibiting a relatively small number of recurrent events, which is the typical situation arising in medical studies. In particular, we study a real dataset containing rehospitalisation times after surgery in patients diagnosed with colorectal cancer, with more covariates.

Keywords: recurrent events; Bayesian nonparametrics; Dirichlet process mixtures; autoregressive processes

Sommario

Questo lavoro propone un nuovo approccio Bayesiano per lo studio di eventi ricorrenti per mezzo di un modello semiparametrico per i tempi di attesa. In particolare, la dipendenza temporale dei tempi di attesa dai precedenti è modellizzata come una mistura di processi autoregressivi. Inoltre, questo modello permette di clusterizzare i pazienti a seconda delle loro traiettorie degli eventi di conteggio nel periodo di osservazione. Nello studio qui presentato è stato possibile introdurre covariate sia fisse che tempo-dipendenti. Come ulteriore contributo originale apportato da questa tesi, si sono ricavate le espressioni analitiche delle full-conditionals, necessarie a costruire un algoritmo MCMC per campionare dalla distribuzione a posteriori dei parametri. L'algoritmo è stato efficientemente implementato usando il linguaggio Julia. I dati di analisi derivano da una grande quantità di processi stocastici, ognuno dei quali produce un numero relativamente piccolo di eventi ricorrenti: questa è la situazione caratteristica degli studi clinici. In particolare, si studia un dataset contenente i tempi di riospedalizzazione post operatoria in pazienti a cui è stato diagnosticato il cancro al colon.

Parole chiave: eventi ricorrenti, Bayesian nonparametrics, Dirichlet process mixtures, modelli autoregressivi

Contents

List of Figures

List of Tables

Introduction

The aim of this work is to propose a new Bayesian semiparametric model to study recurrent event times. Since the literature concerning this particular subject is scarce, two main methodological topics are integrated in the study. On one hand, the main tools for Bayesian non-parametric inference, and in particular the Dirichlet Process Mixture model, are used; on the other hand, the classical framework of survival analysis for recurrent events is taken into account.

The Bayesian approach to recurrent event times has several advantages over its frequentist counterpart. First of all, frequentist inference is based on asymptotic estimates, whereas in Bayesian context the inference is exact even with datasets of small dimension, thanks to numerical integration methods (MCMC). Moreover, when individuals are assumed to be independent in a classical framework, in the corresponding Bayesian model observations are exchangeable, which leads to better estimates. The drawback is that, in general, calculations are more laborious and the implementation is more onerous from a computational point of view.

Stochastic processes that generate events repeatedly over time are referred to as recurrent event processes and the data they provide are called recurrent event data. When individuals frequently experience the events of interest, and the events are "incidental" in the sense that their occurrence does not materially alter the process itself, then methods based on event counts are often employed. Examples of incidental events include mild epileptic seizures or asthmatic attacks in humans.

However, data may also be available for a large number of processes exhibiting a relatively small number of recurrent events. These types of processes arise frequently in medical studies, where information is often available on many individuals, each of whom may experience transient clinical events repeatedly over a period of observation. Examples include myocardial infarctions, severe seizures in epileptic patients, and successive tumours in cancer studies. In this case, if the events are relatively infrequent and the prediction of the time to the next event is of interest, models based on gap times are

used.

This work deals with this second approach to the problem of modelling recurrent events, i.e. gap times between events. In particular, notation and some results presented in Cook and Lawless (2007) are used.

The Bayesian non-parametric approach to recurrent event times has several advantages over the parametric one. In either density estimation or clustering problems a parametric framework could be too restrictive, leading to biased inference and decisions. Instead, it is desirable to consider infinite dimensional families of probability distributions. In this work we propose a mixture model that considers as the mixing random measure a Dirichlet Process. See Müller et al. (2015) for a review of the most common classes of non-parametric priors and of the main Bayesian non-parametric inference techniques.

Let us remark that the posterior inference on a model with a Dirichlet Process prior is carried on infinite unknown parameters. In literature, there are two main schemes to deal with the inference on a infinite-dimensional "parameter", namely marginal and truncation algorithms. The former ones integrate out the infinite dimensional parameter (i.e. the random probability measure), whereas the latter ones approximate the infinite dimensional process with a finite dimensional one. In this work, the first scheme is used. The main advantage of the marginal algorithms is that they are exact, because they do not introduce any truncation error. However, since the random probability measure is integrated out, we cannot recover it and the estimates of the parameters can be obtained by means of the predictive distributions.

The drawback of the Bayesian non-parametric approach consists in its computational heaviness. However, the increasing computational power of the last years has made Bayesian non-parametric inference feasible and more popular in literature. For this reason, the research of efficient algorithms is one of the main goals when dealing with this kind of models. In this work, an efficient Polya urn scheme algorithm is coded in the Julia language, while the post-processing has been developed with the R software.

An additional difficulty in the implementation of the model was its non-conjugacy. In fact, in order to be more flexible with respect to the prior specification and to the choice of the density of the data, the most general case has been implemented following the approach presented with Algorithm 8 in Neal (2000). Once again, the computational heaviness of this algorithm was successfully overcome by an efficient implementation in Julia language, whose performances are similar as the ones of the C language.

In conclusion, the model proposed in this thesis allows for a flexible estimation of

the dependence that each hospitalisation has on the following ones. Since it can be interpreted as a Dirichlet Process Mixture model, it leads to a mixture of autoregressive processes in order to estimate the data density. Clusters of patients are created according to the entire trajectories of the event counts. Moreover, this model allows us to introduce both fixed and time-dependent covariates as another factor of differentiation among patients. The implementation of the model starts from a well-known algorithm (see Neal, 2000). However, the general algorithm has been detailed in this particular case and it has been entirely implemented in an efficient language.

The functioning of the model has been verified on two different simulated datasets. The posterior estimates are able to recover the correct number of clusters and to partition the data coherently with the component of the mixture they were generated from. Subsequently, a study on a real dataset is proposed. In this case, posterior inference on fixed as well as on time-dependent covariates is also available.

The work is organised as follows: in Chapter 1, after a brief introduction to nonparametric Bayesian approach, we present the main properties of the Dirichlet process. The most popular density estimation model, the Dirichlet Process Mixture model, and the sampling strategies for the posterior simulation are discussed. In Chapter 2 the basic theory underlying recurrent event times is presented. A review of existing Bayesian semi-parametric approaches to gap times is proposed. In Chapter 3 we present the new model proposed in this work. Afterwards, the calculation of the full conditionals and the sampling scheme are described. A test on two different simulated datasets is carried out in order to assess the validity of the proposed model and the of the algorithm. In Chapter 4 we present the results of the model on a dataset containing rehospitalisation times after surgery in patients diagnosed with colorectal cancer. Then, the inferences are compared to the ones obtained with the "shared frailty model", which is a semi-parametric method used to estimate the hazard function of the observations. At last, a robustness analysis with respect to the DP prior specification, which is a crucial step in Bayesian nonparametrics, is carried out. In Appendix A we derive the analytic expressions of the full-conditionals of the model, whereas in Appendix B the implementation in the Julia language is briefly described.

Chapter 1

Bayesian Nonparametrics

In this chapter we summarise some relevant results concerning Bayesian nonparametrics, detailing the most popular models and algorithms that are successively used in this work. We refer to Müller et al. (2015) for a full review of Bayesian nonparametrics and its applications.

1.1 The general framework

Classical statistics is based on a framework where observations $X_1, X_2, \dots$ are assumed to be independent and identically distributed (i.i.d.) from an unknown probability distribution P. The statistical problem begins when there exists uncertainty about P. If we denote with p the probability density function (p.d.f.) of P, we say that we are in a parametric framework if p is known to be a member p_θ from a family of distributions $\mathcal{P} = \{p_\theta : \theta \in \Theta\}$, indexed by a finite dimensional parameter θ from a set Θ. The aim of the inference is, in this case, to use the observed sample in order to estimate a plausible value (or a set of values) for θ.

In many situations, however, constraining the analysis to a specific parametric form may be a limit to the inference. Therefore, we would like to relax parametric assumptions in order to allow greater modelling flexibility and robustness against misspecification of a parametric statistical model. In these cases, we may want to consider models where the class of densities is so large that it can no longer be indexed by a finite dimensional parameter, and we therefore require parameters to belong to an infinite dimensional space. We say that we are in a non-parametric framework when P lies in the generic space of probability distributions $\mathcal{P}(\mathbb{R})$.

In Bayesian statistics it is also possible to distinguish between these two alternatives. In the parametric framework we set a prior distribution Π on a finite dimensional space Θ and, given θ, the observations are assumed i.i.d. from P_θ. In the non-parametric

case, we attempt to give a prior Π on the space $\mathcal{P}(\mathbb{R})$ of all probability distributions on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ and, given P, the observations are assumed i.i.d. from P.

Under the assumption of exchangeability, de Finetti's Representation Theorem gives a validation of the Bayesian setting. Let us consider an infinite sequence of observations $\{X_n\}_{n\geq 1}$, with each X_i taking values on $\mathbb{R}$.

Definition 1. A sequence $\{X_n\}_{n\geq 1}$ is exchangeable when, $\forall n \geq 1$ and for any finite permutation π of $(1, 2, \ldots, n)$, the random vectors $(X_1, \ldots, X_n)$ and $(X_{\pi(1)}, \ldots, X_{\pi(n)})$ have the same probability distribution.

This assumption (also called symmetry on the joint law) represents in some way the lack of information. Let us think, for instance, of a sample of binary r.v.s. Then, under the exchangeability assumption, the information in the joint law depends only on the number of 1 and of 0 outcomes, but not on the order of their appearance. This is true in general, i.e. the information that the observations X_i's provide is independent of the order in which they are collected. We also remark that exchangeability implies that the marginal distributions of the X_i's are the same.

Let us now give some preliminary definitions in order to present de Finetti's Representation Theorem. Let $\mathcal{P}(\mathbb{R})$ be the space of all probability measures on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ and let $\mathcal{C}_{\mathcal{P}}$ be the Borel σ-algebra on $\mathcal{P}(\mathbb{R})$. This latter is the the smallest σ-algebra generated by the open sets in the weak topology, i.e. the smallest σ-algebra that makes the sets $\{P \in \mathcal{P}(\mathbb{R}) : P(B) \leq t\}$ measurable $\forall t \in [0, 1]$, $\forall B \in \mathcal{B}(\mathbb{R})$. For further details about the notions of weak topology and weak convergence, see Regazzini (1996).

Definition 2. A random probability measure (r.p.m.) is a random element $P : (\Omega, \mathcal{F}, \mathbb{P}) \to (\mathcal{P}(\mathbb{R}), \mathcal{C}_{\mathcal{P}})$, i.e. it is a stochastic process whose trajectories $\omega \to P(\omega)$ are probability measures on $\mathbb{R}$.

In the Bayesian context, a r.p.m. will be given as the conditional distribution of the observations $P \sim \pi(\cdot)$. For application purposes, there are two desirable properties for r.p.m.s. First of all, we require a full support for the prior distribution, i.e. $supp(\pi) = \mathcal{P}(\mathbb{R})$, in order to explore the space of all possible probability distributions. Moreover, the posterior has to be analytically tractable in order to lead to a computationally feasible model. The latter property is now less necessary thanks to the development of Markov Chain Monte Carlo (MCMC) methods.

Theorem 1 (de Finetti)**.** *Let $\{X_n\}_{n\geq 1}$ be a sequence of r.v.'s with values in $\mathbb{R}$. Then, $\{X_n\}_{n\geq 1}$ is exchangeable if and only if there exists a unique r.p.m. $P : (\Omega, \mathcal{F}) \to$*

$(\mathcal{P}(\mathbb{R}), \mathcal{C}_{\mathcal{P}})$ *such that*

$$\mathbb{P}(X_1 \in A_1, \ldots, X_n \in A_n) = \int_{\mathcal{P}(\mathbb{R})} \prod_{i=1}^{n} P(A_i)\pi(dP) \quad \forall n \geq 1, \quad \forall A_i \in \mathcal{B}(\mathbb{R}).$$

In other words, for any $n = 1, 2, \ldots$

$$X_1, \ldots, X_n | P \overset{\text{iid}}{\sim} P$$
$$P \sim \pi(\cdot).$$

This theorem justifies the Bayesian approach, since the existence of a r.p.m. as the conditional distribution of the data is implied by the mild assumption of exchangeability.

In the non-parametric Bayesian context, inference and prediction are carried out similarly to the parametric case. If we denote with π a probability measure on $\mathcal{P}(\mathbb{R})$, the posterior distribution can be expressed by means of Bayes' Theorem, i.e.

$$\mathcal{L}(dP|X_1, \ldots, X_n) = \frac{\prod_{i=1}^{n} P(x_i)\pi(dP)}{\int_{\mathcal{P}(\mathbb{R})} \prod_{i=1}^{n} P(x_i)\pi(dP)}$$

and the predictive distribution of a new observation X_{n+1} is

$$\mathbb{P}(X_{n+1} \in A | X_1, \ldots, X_n) = \int_{\mathcal{P}(\mathbb{R})} P(A)\mathcal{L}(dP|X_1, \ldots, X_n).$$

1.2 Dirichlet Process

In this section, the Dirichlet process is defined and its properties are presented. All the definitions are given over $\mathbb{R}$, but any generalisation over $\mathbb{R}^p$ with $p \geq 1$ is evident.

1.2.1 Definition and Properties

The Dirichlet Process (DP) prior is one of the most popular families of BNP models. Originally introduced by Ferguson (1973), the DP is a prior on the space of probability measures $\mathcal{P}(\mathbb{R})$. It is straightforward to define it from its finite-dimensional analogous (the Dirichlet Distribution) and it has some nice properties, such as the conjugacy.

First of all, let us recall the definition of the Dirichlet distribution.

Definition 3. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ with $\alpha_i \geq 0 \ \forall i = 1, \ldots, k$. The random vector $\boldsymbol{P} = (P_1, \ldots, P_k)$, $\sum_{i=1}^{k} P_i = 1$, $P_i \geq 0 \ \forall i = 1, \ldots, k$, has Dirichlet distribution with parameter $\boldsymbol{\alpha}$ if $(P_1, \ldots, P_{k-1})$ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^{k-1}$ with density

$$f(p_1, \ldots, p_{k-1}) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_k)}{\Gamma(\alpha_1) \ldots \Gamma(\alpha_k)} p_1^{\alpha_1 - 1} \ldots p_{k-1}^{\alpha_{k-1} - 1} \left(1 - \sum_{i=1}^{k-1} p_i\right)^{\alpha_k - 1} \mathcal{I}_{\mathcal{S}_{k-1}}(p_1, \ldots, p_{k-1}),$$

where $\mathcal{I}_{\mathcal{S}_{k-1}}(p_1, \ldots, p_{k-1})$ is the indicator function on the $k-1$-dimensional simplex, defined as the set $\mathcal{S}_{k-1} = \left\{(p_1, \ldots, p_{k-1})' \in \mathbb{R}^{k+1} : \sum_{i=1}^{k-1} p_i = 1,\ p_i \geq 0\ \forall i\right\}$. We will write $\boldsymbol{P} \sim \text{Dirichlet}(\boldsymbol{\alpha})$.

By generalising to the infinite-dimensional case, we introduce the following definition (Ferguson, 1973).

Definition 4. Let α be a finite measure on $\mathbb{R}$, and $M \coloneqq \alpha(\mathbb{R})$; let $P_0(\cdot) = \alpha(\cdot)/M$. A r.p.m. $P : (\Omega, \mathcal{F}, \mathbb{P}) \to (\mathcal{P}(\mathbb{R}), \mathcal{C}_{\mathcal{P}})$ is a Dirichlet Process if, for any finite measurable partition $A_1, \ldots, A_m$ of $\mathbb{R}$,

$$(P(A_1), P(A_2), \ldots, P(A_m)) \sim \text{Dirichlet}(\alpha(A_1), \alpha(A_2) \ldots, \alpha(A_m)).$$

We will write $P \sim \text{DP}(MP_0)$.

The collection of finite dimensional distributions implies a well defined (existent and unique) process with values on $\mathcal{P}(\mathbb{R})$; see Ferguson (1973) for the proof. The parameter M is called the precision or total mass parameter, P_0 is the centering measure, and the product $M \times P_0$ is referred to as the base measure of the DP. If $P \sim \text{DP}(MP_0)$, it follows that $\mathbb{E}[P(A)] = P_0(A)$ for any Borel set A, and thus we can interpret P_0 as the prior expectation of P. Moreover, $\text{Var}[P(A)] = \frac{P_0(A)(1-P_0(A))}{M+1}$, therefore the total mass of the DP reflects the degree of belief in the prior expectation.

One of the most important properties of the Dirichlet process prior is its conjugacy. In fact, let $(X_1, \ldots, X_n)$ be a sample from a Dirichlet process, i.e.

$$\begin{aligned} X_1, \ldots, X_n | P &\overset{\text{iid}}{\sim} P \\ P &\sim \text{DP}(MP_0). \end{aligned}$$

Then, the posterior distribution is given by

$$P | X_1, \ldots, X_n \sim \text{DP}(MP_0 + \sum_{i=1}^{n} \delta_{X_i}).$$

We remark that, $\forall A \in \mathcal{B}(\mathbb{R})$, the posterior mean

$$\mathbb{E}[P(A) | X_1, \ldots, X_n] = \frac{M}{M+n} P_0(A) + \frac{n}{M+n} \frac{\sum_{i=1}^{n} \delta_{X_i}(A)}{n}$$

is a weighted sum of the prior expectation and of the empirical distribution of $\{X_1, \ldots, X_n\}$.

It can also be proved that the predictive distribution of X_{n+1} has the following representation:

$$\begin{aligned} X_1 &\sim P_0 \\ X_{n+1} | X_1, \ldots, X_n &\sim \frac{M}{M+n} P_0 + \frac{n}{M+n} \frac{\sum_{i=1}^{n} \delta_{X_i}}{n}. \end{aligned} \tag{1.1}$$

The predictive distribution (1.1), also called *Blackwell-Macqueen Urn Scheme* (see Pitman, 1996, for further details), implies that a new value is sampled either from a baseline measure P_0 with probability $\frac{M}{M+n}$ or from one of the previous sampled values, each one with probability $\frac{1}{M+n}$. Therefore there is a positive probability of obtaining ties in the sample. The allocation process associated with the predictive distribution is also known as the *generalised Polya urn*. Consider an urn that initially has M black balls and one coloured ball (whose "colour" is randomly selected according to P_0). We sequentially draw balls from the urn; if a coloured ball is drawn then we return it to the urn along with another ball of the same color; if a black ball is drawn, we return it to the urn along with a ball of a new color randomly selected according to P_0.

Formula (1.1) also allows us to sample P without simulating any trajectory of the Dirichlet process, which will be a fundamental feature for the Polya urn scheme algorithm used in this thesis.

1.2.2 Sethuraman's construction

Let us now give a constructive definition of the Dirichlet process. Let us consider two independent sequences of r.v.s, $\{\theta_i\}_{i\geq 1}$ and $\{v_i\}_{i\geq 1}$ s.t. $\theta_i \overset{\text{iid}}{\sim} P_0$ and $v_i \overset{\text{iid}}{\sim} \text{Beta}(1, M)$. Let us now define the weights

$$\begin{cases} w_1 = v_1 \\ w_i = v_i \prod_{j=1}^{i-1}(1 - v_j). \end{cases}$$

Sethuraman (1994) proved that

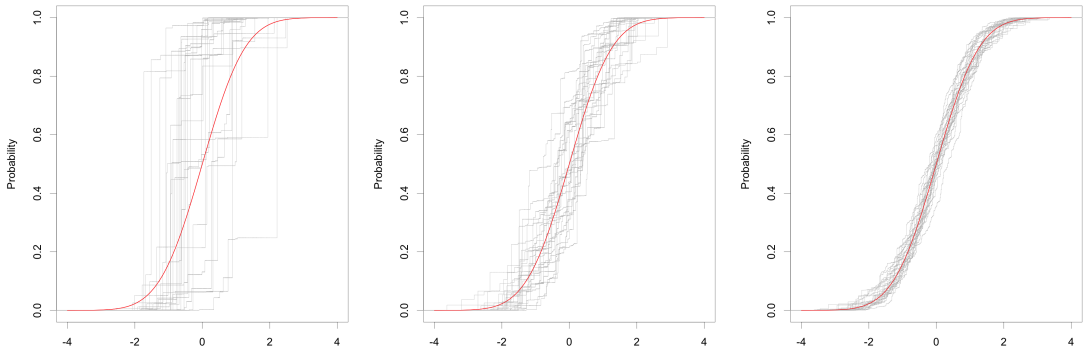
$$P(\cdot) \overset{\text{d}}{=} \sum_{n=1}^{\infty} w_n \delta_{\theta_n}(\cdot) \tag{1.2}$$

where w_i and θ_i are defined above.

Equation (1.2) is called *stick-breaking representation* because of the analogy with a stick of unit length: w_1 represents a piece of the stick, w_2 a piece of the remainder obtained after cutting w_1 away, and so on. Each piece is independently modelled as a $\text{Beta}(1, M)$ random variable scaled to the length of the remainder of the stick.

From this construction it is clear that the Dirichlet process has discrete trajectories even if P_0 is continuous, i.e. if $P \sim \text{DP}(MP_0)$, then $\mathbb{P}(\{\omega : P(\omega) \text{ is discrete}\}) = 1$. Moreover, this useful construction also allows an easy visualisation of the trajectories of a Dirichlet process, as seen in Figure 1.1.

Let E be the support of the finite measure P_0 on $\mathbb{R}$. Then it can be shown that the weak support of the Dirichlet process is $supp(\text{DP}(MP_0)) = \{P \in \mathcal{P} : supp(P) \subset E\}$, i.e.



(a) M=1 *(b) M=10* *(c) M=100*

Figure 1.1: Plots of 25 *samples from a DP(MP_0), with $P_0 = \mathcal{N}(0,1)$ for three different values of M. P_0 is overlaid with a red line.*

the set of all the probability distributions with support contained in the support of the measure P_0 is the weak support of $\mathrm{DP}(MP_0)$. In particular, if we assume that $E = \mathbb{R}$ (for example a Gaussian distribution), then the Dirichlet process has full support, i.e. $supp(\mathrm{DP}(MP_0)) = \mathcal{P}(\mathbb{R})$.

1.2.3 Data clustering and density estimation

If we consider a sample $(X_1, X_2, \ldots, X_n)$ from a Dirichlet process P, where $P \sim \mathrm{DP}(MP_0)$, we saw in formula (1.1) that some values are coincident with positive probability. Thus, a partition of the indexes $\{1, 2, \ldots, n\}$, denoted with $\rho_n = \{S_1, \ldots, S_k\}$, and therefore a clustering structure are induced by the ties in the sample, where k is the number of unique values in the sample and $S_j = \{i \in \{1, \ldots, n\} \text{ s.t. } X_i = X_j\}$ are the indexes of the j^{th} group. Let us denote with $\boldsymbol{n} = (n_1, \ldots, n_k)$ the cluster sizes for a partition of n observations into clusters $S_1, \ldots, S_k$. The prior distribution induced on ρ_n by $(X_1, \ldots, X_n)$ is in this case

$$\pi(\rho_n; , n_1, \ldots, n_k) = \frac{\Gamma(M)}{\Gamma(M+n)} M^k \prod_{j=1}^{k} \Gamma(n_j),$$

where $\boldsymbol{n}$ is such that $\sum_{i=1}^{k} n_i = n$ and $k = 1, 2, \ldots, n$. From this prior distribution, also known as exchangeable product partition function (EPPF), Antoniak (1974) proved that the marginal law of the prior number of clusters is

$$\pi(K_n = k) = |S_1(n,k)| M^k \frac{\Gamma(M)}{\Gamma(M+n)}, \tag{1.3}$$

where $S_1(n,k)$ is the Stirling number of the first kind, which can be tabulated or computed by a software. From (1.3) one can remark the influence of the mass parameter M

on the number of clusters: a larger value of M gives rise to a higher prior number of components.

The DP model presented above, albeit useful for clustering purposes, has a serious limitation if used in order to estimate the density of a population, which is one of the main goals of Bayesian nonparametrics. In fact, it can be shown that

$$\mathbb{E}[P|X_1, \ldots, X_n] = \mathcal{L}(X_{n+1}|X_n, \ldots, X_1),$$

and therefore there are ties in the posterior distribution P, which is nonsense when estimating continuous distributions.

1.3 Dirichlet Process Mixture

The most popular r.p.m.s in literature are Dirichlet Processes and Polya Trees. See Müller et al. (2015) for a recent review of the main r.p.m. classes. The purpose of this section is to present the DPM model, which is useful for our purposes of density estimation and regression.

1.3.1 The model

As mentioned before, the discrete nature of the DP random measure is awkward when the unknown distribution is known to be continuous. One way to fix this limitation of the DP model is to consider instead a mixture of a continuous kernel with respect to the discrete distribution P. This approach has been widely studied by Escobar and West (1995).

Let Θ be a finite dimensional parameter space, and let $\{k(x;\theta), \theta \in \Theta\}$ be a family of parametric probability distributions, i.e.

$$\begin{aligned} &x \mapsto k(x;\theta) \text{ is a density } \forall \theta \in \Theta \\ &\theta \mapsto k(x;\theta) \text{ is a measurable function } \forall x \in \mathbb{R}. \end{aligned}$$

We say that $X_1, \ldots, X_n$ is a sample from the Dirichlet Process Mixture model (DPM) when

$$\begin{aligned} &X_i|P \overset{\text{iid}}{\sim} f(x) = \int_\Theta k(x;\theta)P(d\theta), \\ &P \sim \text{DP}(MP_0). \end{aligned} \tag{1.4}$$

Note that $f(x)$ in (1.4) is a random density, because P is a random probability measure, i.e. $f(x;\omega) = \int_\Theta k(x;\theta)P(d\theta;\omega)$.

By exploiting Sethuraman's construction (1.2) of the mixing r.p.m. and plugging it in (1.4) we easily obtain

$$\begin{aligned} X_i|P \overset{\text{iid}}{\sim} f(x)(\omega) &= \int_\Theta k(x;\theta) \sum_{j=1}^{\infty} w_j(\omega)\delta_{\theta_j(\omega)}(d\theta) \\ &= \sum_{j=1}^{\infty} w_j(\omega) k(x;\theta_j(\omega)). \end{aligned}$$

Therefore, the population density $f(x)(\omega)$ is the mixture of infinitely many parametric distributions, where the DP random measure is the mixing measure.

The mixture model (1.4) can equivalently be written as a hierarchical model as follows

$$\begin{aligned} &X_i|\theta_i \overset{\text{ind}}{\sim} k(\cdot;\theta_i) && i = 1,\dots,n \\ &\theta_i|P \overset{\text{iid}}{\sim} P && i = 1,\dots,n \\ &P \sim \text{DP}(MP_0). \end{aligned} \tag{1.5}$$

The representation in (1.5) is more useful for our purposes: the parameters θ_i are a sample from a DP, therefore some of them are coincident with positive probability. Since each observation X_i is associated to a latent parameter θ_i, the clustering structure will be based on the ties in the sample of the θ_i's.

Under this hierarchical model, the posterior distribution on P is itself a mixture of DP, i.e.

$$\mathcal{L}(P|x_1,\dots,x_n) \sim \int_{\Theta^n} \text{DP}(MP_0 + \sum_{i=1}^{n} \delta_{\theta_i}) \mathcal{H}(d\theta_1,\dots,d\theta_n|x_1,\dots,x_n).$$

However, unless the model is conjugate, there is no simple strategy in order to sample from the posterior distribution of the latent parameters $\mathcal{H}(d\boldsymbol{\theta}|\boldsymbol{X})$. The main algorithms present in literature are reviewed in Section 1.3.2.

1.3.2 Algorithms

The aim of this section is to present a review of the MCMC algorithms for fitting models with DP priors. For a full and exhaustive dissertation on the algorithms and on their implementation schemes, see Neal (2000).

The models we deal with, in the most frequent cases, do not admit conjugate priors and can have a very complicated structure. In such cases, the calculation of the posterior distribution of the parameters is not straightforward. When the posterior distribution is not exploitable in an easy way, we recur to numerical simulation. Markov Chain

Monte Carlo (MCMC) methods are techniques which allow the numerical evaluation of the posterior density even when one cannot find a closed form for it.

We are typically interested in the expected value of a function of the unknown parameter θ, say $h(\theta)$, i.e.

$$\mathbb{E}_\pi[h(\theta)|\text{data}] = \int_\Theta h(\theta)\pi(d\theta|\text{data}).$$

The strong law of large numbers states that, given $\{\theta^{(g)}\}_{g=1}^G$ sequence of independent draws from the density $\pi(d\theta|\text{data})$, then

$$\mathbb{E}_\pi[h(\theta)|\text{data}] = \frac{1}{G}\sum_{i=1}^{G} h(\theta^{(i)}). \tag{1.6}$$

This results holds even in the case we relax the assumption of independent draws. In particular, let us suppose we can sample a Markov Chain with values in Θ that satisfies some properties (Harris-recurrence and irreducibility), such that its limit distribution is the target distribution $\pi(\theta|\text{data})$. Then one can estimate the function $h(\theta)$ of the unknown parameter as in (1.6). From a practical point of view, then, the only problem is to generate a sample of autocorrelated values (i.e. a Markov Chain) whose invariant distribution is $\pi(\theta|\text{data})$. Gibbs Sampler and Metropolis-Hastings are algorithms that will be useful for this purpose. For more details about Markov Chains, see Jackman (2009).

In the case of models with DP priors, the algorithms are divided into two main classes, according to their computational strategy: the first group consists of schemes marginalising out the random probability measure P (collapsed Gibbs Samplers); the second group of all the algorithms which impute the Dirichlet process and update it as a component of the Gibbs sampler (conditional methods). In the latter group, some algorithms exploit the stick breaking representation of P (blocked Gibbs Samplers), see Ishwaran and James (2001), whereas others rely on the technique of retrospective sampling, see Papaspiliopoulos and Roberts (2008). In this work, a particular case of collapsed Gibbs Sampler will be presented and used.

Collapsed Gibbs Samplers

For this kind of algorithms, the Polya urn representation of the predictive distribution of a sample from a DP prior is the fundamental ingredient. In fact, let us suppose we have

a model as in (1.5); then one can prove (see Escobar and West, 1995) that

$$\begin{aligned}\mathcal{L}(X_{n+1}|X_n,\ldots,X_1) &= \int_{\mathcal{P}(\mathbb{R})} \mathcal{L}(X_{n+1}, dP|X_1,\ldots,X_n) \\ &= \int_{\Theta^n} \left(\frac{M}{M+n} q_0(x) + \frac{1}{M+n} \sum_{j=1}^{n} k(x;\theta_j) \right) \mathcal{H}(d\theta_1,\ldots,d\theta_n|\boldsymbol{X}),\end{aligned}$$

where

$$q_0(x) = \int_{\Theta} k(x;\theta_{n+1}) P_0(d\theta_{n+1}) \tag{1.7}$$

admits a simple form only if the model is conjugate. In this latter case, it is not hard to write the full conditionals of a Gibbs sampler for the posterior distribution of the latent parameters $\theta_1,\ldots,\theta_n$. Therefore, given a posterior sample $\boldsymbol{\theta}^{(1)},\ldots,\boldsymbol{\theta}^{(G)}$, one can compute the density

$$f(x|\boldsymbol{X}) = \frac{M}{M+n} q_0(x) + \frac{1}{M+n} \left[\frac{1}{G} \sum_{g=1}^{G} k(x;\theta_j^{(g)}) \right].$$

The first strategy to sample from the posterior of the latent variables θ_i was proposed by Escobar and West (1995). This version of the algorithm is based on transition probabilities that update θ_i by draws from the complete conditional posterior $\mathcal{L}(\theta_i|\boldsymbol{X},\boldsymbol{\theta}_{-i})$. However, this Gibbs sampler suffers from a slowly mixing. Therefore we directly present here a variation of this algorithm, first presented by Bush and MacEachern (1996), which introduces an acceleration step.

Two vectorial quantities are introduced: $\boldsymbol{\theta}^* = (\theta_1^*,\ldots,\theta_k^*)$ is the vector of distinct entries in $\boldsymbol{\theta}$; $\boldsymbol{S} = (s_1,\ldots,s_n)$ is the vector of labels associated to each observation, i.e. $s_i = j \Leftrightarrow \theta_i = \theta_j^*$. Therefore, the information provided by $\boldsymbol{\theta}$ is the same conveyed in the joint vector $(\boldsymbol{\theta}^*, \boldsymbol{S})$. Two transition probabilities are needed in order to update both. One updates s_i by draws from the complete conditional posterior probability $\mathcal{L}(s_i|\boldsymbol{s}_{-i},\boldsymbol{X})$, after marginalizing with respect to $\boldsymbol{\theta}$ (we denoted with the subscript $-i$ the vector without the element i). The other type of transition probability samples from $\mathcal{L}(\theta_j^*|\boldsymbol{s},\boldsymbol{X})$.

The probabilities $\mathcal{L}(s_i|\boldsymbol{s}_{-i},\boldsymbol{X})$ are derived as follows. Let us first consider $\mathcal{L}(\theta_i|\boldsymbol{\theta}_{-i})$, which is the predictive distribution of a sample from a DP prior. As we have seen in (1.1), this expression is equivalent to

$$\theta_i|\boldsymbol{\theta}_{-i} \sim \frac{M}{M+n-1} P_0 + \frac{n-1}{M+n-1} \frac{\sum_{j=1,j\neq i}^{n-1} \delta_{\theta_j}}{n-1}. \tag{1.8}$$

Recall that $\theta_j^{*(-i)}$ denotes the $k^{(-i)}$ unique values among $\boldsymbol{\theta}_{-i}$, and $n_j^{(-i)}$ the cluster sizes in the vector $\boldsymbol{\theta}_{-i}$. By multiplying equation (1.8) by the sampling distribution $\mathcal{L}(\boldsymbol{X}|\boldsymbol{\theta})$,

after normalisation we obtain

$$\mathcal{L}(d\theta_i|\boldsymbol{\theta}_{-i},\boldsymbol{X}) = \frac{Mk(x_i;\theta_i)P_0(d\theta_i)}{Mq_0(x_i)+\sum_{j\in\boldsymbol{s}_{-i}} n_j^{(-i)}k(x_i;\theta_j^{*(-i)})} + \frac{\sum_{j\in\boldsymbol{s}_{-i}} n_j^{(-i)}k(x_i;\theta_j^{*(-i)})\delta_{\theta_j^{*(-i)}}(d\theta_i)}{Mq_0(x_i)+\sum_{j\in\boldsymbol{s}_{-i}} n_j^{(-i)}k(x_i;\theta_j^{*(-i)})},$$

where $q_0(\cdot)$ is defined in (1.7). This expression can be rewritten as the distribution of (θ_i, s_i) by simply remarking that $\theta_i = \theta_j^{*(-i)}$ implies $s_i = j$. Therefore, after marginalising with respect to $\boldsymbol{\theta}$, we get the desired full conditional.

As far as the probability $\mathcal{L}(\theta_j^*|\boldsymbol{s},\boldsymbol{X})$ is concerned, we update the cluster specific parameters conditioning on the imputed partition $\boldsymbol{s}$ using

$$\mathcal{L}(\theta_j^*|\boldsymbol{s},\boldsymbol{X}) \propto P_0(\theta_j^*)\prod_{i:s_i=j} k(x_i;\theta_j^*). \tag{1.9}$$

In this work a different approach will be used. Since the algorithm presented above is only feasible when conjugate prior P_0 are used (otherwise the exact computation of $q_0(\cdot)$ is not analytically tractable), we need a more general framework to set our model.

Neal's Algorithm 8

MacEachern and Müller (1998) propose the "no-gaps" algorithm, that does allow auxiliary values for θ drawn from P_0 to be used to define a valid Markov chain sampler. As noted by Neal (2000), however, this algorithm is inefficient when creating new clusters and when assigning an observation to a newly created mixture component. The probability of such a change, indeed, is reduced from what one might expect by a factor of $k^{(-i)}-1$.

We here discuss a variation of this algorithm, Algorithm 8 in Neal (2000), which overcomes this issue. Moreover, this latter version introduces auxiliary variables in the MCMC scheme in order to evaluate via Monte Carlo the integral involved in the calculation of the marginal $q_0(\cdot)$. The data augmentation consists in substituting the base measure P_0 with the empirical distribution of a random sample of size m from P_0 itself.

Let $\boldsymbol{\psi}^{(i)} = (\psi_1^{(i)},\dots,\psi_m^{(i)}) \quad \forall i = 1,\dots,n$ an i.i.d. sample of size m from P_0. Then

$$\mathcal{L}(d\theta_1|\boldsymbol{\psi}^{(1)}) \sim \frac{1}{m}\sum_{j=1}^{m}\delta_{\psi_j^{(1)}}(d\theta_1)$$

$$\mathcal{L}(d\theta_i|\theta_1,\dots,\theta_{i-1},\boldsymbol{\psi}^{(i)}) \sim \frac{M/m}{M+i-1}\sum_{j=1}^{m}\delta_{\psi_j^{(i)}}(d\theta_i) + \frac{i-1}{M+i-1}\sum_{h<i}\delta_{\theta_h}(d\theta_i), \quad \forall i = 2,\dots,n,$$

which is the Polya urn scheme (1.1) where we have replaced a single value sampled from P_0 with a value chosen at random from a sample of size m of P_0.

In analogy with Argiento et al. (2009), the augmenting variables are not discarded at each iteration but they reside in the state space $(\boldsymbol{\theta}, \boldsymbol{s}, \boldsymbol{\psi}^{(1)}, \ldots, \boldsymbol{\psi}^{(n)})$. The Gibbs sampler is organised in a sequential fashion: first, the labels are updated and the corresponding $\boldsymbol{\theta}$ parameters are exchanged, i.e. the vector (θ_i, s_i) is updated; second, the unique $\boldsymbol{\theta}^*$ parameters are updated using the information given by the observations in each cluster.

The full conditionals of the first transition kernel are the following:

$$\mathbb{P}(s_i^{'} = j | \boldsymbol{s}_{-i}, \boldsymbol{\theta}, \boldsymbol{\psi}^{(1)}, \ldots, \boldsymbol{\psi}^{(n)}, \text{data}) \propto \begin{cases} k(x_i; \theta_j^*) n_j^{(-i)} & \text{for } j \in \boldsymbol{s}_{-i} \\ \frac{M}{m} \sum_{h=1}^{m} k(x_i; \psi_h^{(i)}) & \text{for } j = k_{new} \end{cases}$$

$$\mathbb{P}(d\theta_i^{'} | \boldsymbol{s}^{'}, \boldsymbol{\theta}_{-i}, \boldsymbol{\psi}^{(1)}, \ldots, \boldsymbol{\psi}^{(n)}, \text{data}) \propto \begin{cases} \delta_{\theta^*_{s_i^{'}}}(d\theta_i^{'}) & \text{if } s_i^{'} \in \boldsymbol{s}_{-i} \\ \sum_{h=1}^{m} k(x_i; \psi_h^{(i)}) \delta_{\psi_h^{(i)}}(d\theta_i^{'}) & \text{if } s_i^{'} = k_{new} \end{cases}$$

$$\mathbb{P}(d\boldsymbol{\psi}^{(i)} | \boldsymbol{s}^{'}, \boldsymbol{\theta}^{'}, \text{data}) = \begin{cases} \prod_{h=1}^{m} P_0(d\psi_h^{(i)}) & \text{if } s_i^{'} \in \boldsymbol{s}_{-i} \\ \delta_{\theta_i^{'}}(d\psi_{\overline{h}}^{(i)}) \prod_{h \neq \overline{h}} P_0(d\psi_h^{(i)}) & \text{if } s_i^{'} = k_{new}, \end{cases}$$

where $\psi_{\overline{h}}^{(i)}$ is the element of $\boldsymbol{\psi}^{(i)}$ that was assigned to $\theta_i^{'}$. The unique $\boldsymbol{\theta}^*$ parameters are updated according to (1.9), and in this case a step of Metropolis within Gibbs is required.

For further details about the algorithm and its implementation, see Appendix B.

Chapter 2

Event histories and recurrent events

Inference for survival analysis is one of the traditional applications of non-parametric Bayesian inference. In this chapter, event time data are presented and non-parametric approaches applied to recurrent events are discussed.

2.1 Survival analysis

In this section we consider the analysis of time-to-event data. We mention Christensen et al. (2011) for a comprehensive presentation of the main models and methods.

Survival analysis is the term used to describe the analysis of time-to-event data in biological and medical contexts. Reliability analysis is often used for non-biological applications. Examples of this kind of data include: (i) the time until death after diagnosis with leukaemia; (ii) the time it takes to get sick after infection with a virus; (iii) the time until a machine breaks down after being installed.

The most common goal of survival analysis is to compare survival prospects among different populations. Let T denote the random variable representing the survival times of individuals in some population. Time-to-event data are distinguished by two features:

- they are positive, i.e. T is a non-negative random variable. Moreover, time-to-event data are often skewed so we would need to take a log transformation before analysing them;

- they are often censored (i.e. partially observed). We often know that a unit (a person, a machine) was operative (alive, working) up to a certain time but do not know exactly when it failed or would fail.

In this work, we consider time-to-event data as continuous random variables.

Let $f(t)$ denote the probability density function (p.d.f.) of T and let $F(t) = \mathbb{P}(T \leq t)$ be the cumulative distribution function of T. With time-to-event data, the primary object of analysis is the survival function $S(t)$, defined as $S(t) = 1 - F(t) = \mathbb{P}(T > t)$.

The hazard function $h(t)$ is the instantaneous rate of failure at time t and is defined by

$$h(t) = \lim_{\Delta t \to 0} \frac{\mathbb{P}(t \leq T \leq t + \Delta t | T > t)}{\Delta t} = \frac{f(t)}{1 - F(t)} = \frac{f(t)}{S(t)}.$$

In particular, $h(t)\Delta t$ is the approximate probability of failure in $(t, t + \Delta t)$ given survival up to time t.

The functions $f(t)$, $F(t)$, $S(t)$, and $h(t)$ give mathematically equivalent specifications of the distribution of T. In particular, one can prove that

$$S(t) = \exp\left(-\int_0^t h(u)du\right).$$

As previously mentioned, data are not always completely observable. For example, when executing life tests, one cannot wait for all units to break down. For some units, only the survival up to time t^* is observed; this mechanism is called right censoring, i.e. we may only know that the event will happen later than t^*. Let C be a random variable that denotes the time at which a censoring mechanism kicks in. What we actually observe in time-to-event studies is either the event time T or the censoring time C, whichever is smaller. The observed data for each sample unit are

$$y = \min\{T; C\}$$

and

$$\delta = \begin{cases} 1 & \text{if } T \leq C \\ 0 & \text{if } T > C. \end{cases}$$

To simplify the study, two assumptions are necessary:

- T and C are independent;
- the censoring distribution, say $G(c) = \mathbb{P}(C \leq c)$, does not depend on any of the same parameters as $S(t)$ (uninformative censoring).

In this case, if we also have independent observations (y_i, δ_i) $i = 1, \ldots, n$, the likelihood of the data can be expressed simply as the product of the densities for all of the actual observed survival times multiplied by the product of the probabilities for all of the censored observations, i.e.

$$L(\theta; \text{data}) \propto \prod_{i=1}^{n} [f_i(y_i|\theta)]^{\delta_i} [S_i(y_i|\theta)]^{1-\delta_i},$$

which can be rewritten, in terms of the hazard function, as

$$L(\theta; \text{data}) \propto \prod_{i=1}^{n} [h_i(y_i|\theta)]^{\delta_i} [S_i(y_i|\theta)].$$

In practice, the survival distribution and the density are unknown and we need to estimate them from data. We can either assume that the survival distribution belongs to a parametric family, e.g., log-normal, Exponential, Weibull, or Gamma, or we can take a non-parametric approach to estimate the survival curve. If covariates are available in the study, time-to-event analysis can be generalised via, for example, the accelerated failure time (AFT) model or the proportional hazards (PH) model. These two popular models rely on different hypothesis. Moreover, the first is fully parametric, whereas the latter can be specified in a semiparametric fashion and is therefore more useful when dealing with multimodal distributions.

For the purposes of this work, it is necessary to introduce also some techniques that enhance the analysis of data with more than one event per unit.

2.2 Recurrent events and gap times

As we discussed, in classical survival analysis one focuses on a single event for each individual, describing the occurrence of the event by means of survival curves or hazard rates and analysing the dependence on covariates by means of regression models. The connection of several events (of the same kind) for an individual as they occur over time yields to the main subject of this chapter, i.e. event histories.

Processes that generate events repeatedly over time are referred to as recurrent event processes and the data they provide are called recurrent event data. Data may be available for a large number of processes (patients) exhibiting a relatively small number of recurrent events. These types of processes arise frequently in medical studies, where information is often available on many individuals, each of whom may experience transient clinical events repeatedly over a period of observation. Examples include myocardial infarctions, seizures in epileptic patients, and successive tumours in cancer studies.

For a single recurrent event process, which is a point process, starting for simplicity at $t = 0$, let $0 \leq T_1 < T_2 < \dots$ denote the event times, where T_k is the time of the k^{th} event. The associated counting process $\{N(t), 0 \leq t\}$ records the cumulative number of events generated by the process; specifically, $N(t) = \sum_{k=1}^{\infty} \mathcal{I}_{\{T_k \leq t\}}$ is the number of events occurring over the time interval $[0, t]$. More generally, $N(s, t) = N(t) - N(s)$ represents the number of events occurring over the interval $(s, t]$. As defined here, counting processes are right-continuous, that is, $N(t) = N(t^+)$.

There exist two different approaches to event occurrences: event counts or gap times between successive events.

The first approach uses results from stochastic calculus and point processes. Methods based on counts are often useful when individuals frequently experience the events of interest, and the events are "incidental" in the sense that their occurrence does not materially alter the process itself. Models of this kind can be specified very generally by considering the probability distribution for the number of events in short intervals $[t, t + \Delta t)$. For events occurring in continuous time we make the mathematically convenient assumption that two events cannot occur simultaneously. Then, the intensity process $\lambda(t)$ is defined as the conditional probability that an event occurs in $[t, t + \Delta t)$, given all that has been observed prior to this interval, divided by the length of the interval. More formally

$$\lambda(t|\mathcal{H}(t)) = \lim_{\Delta t \to 0} \frac{\mathbb{P}(\Delta N(t) = 1|\mathcal{H}(t))}{\Delta t},$$

where $\Delta N(t) = N(t+\Delta t^-) - N(t^-)$ denotes the number of events in the interval $[t, t+\Delta t)$, and $\mathcal{H}(t) = \{N(s) : 0 \leq s < t\}$ denotes the history of the process at time t. For a full review of this class of methods, see Aalen et al. (2008).

In this work the second approach to recurrent events is used, i.e. modelling gap times between successive events. Analyses based on waiting times are often useful when events are relatively infrequent, when some type of individual renewal occurs after an event, or when prediction of the time to the next event is of interest. We follow here the notation and the general framework of Cook and Lawless (2007).

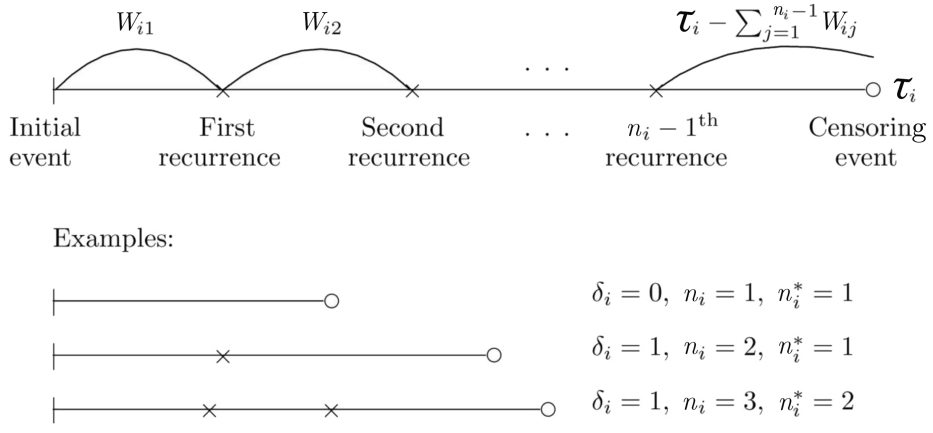


Figure 2.1: Representation of recurrent events for a generic observation. We denote with δ_i the censoring indicator, with n_i the total number of recurrent events and with n_i^* the number of observed recurrent events (without censoring).

Let W_{ij}, $j = 1, \ldots, n_i$ denote waiting times (or gap times) between the $(j-1)^{st}$ and j^{th} event of patient i. Let us assume that $t = 0$ corresponds at the start of each event process and that individual i is observed over the time interval $[0, \tau_i]$. By $\boldsymbol{x}_{ij}$ we denote a vector of possibly time-varying covariates at time j. If n_i events are observed at times $0 < t_{i1} < \cdots < t_{in_i} \leq \tau_i$, let $w_{ij} = t_{ij} - t_{ij-1}$ for $j = 1, \ldots, n_i$ and $w_{i,n_i+1} = \tau_i - t_{in_i}$ where $t_{i0} = 0$. These are the observed gap times for individual i with the final time being possibly censored. Let J denote the maximum number of observed repeated events, i.e. $J = \max_{i=1,\ldots,n} n_i$. A visual representation of such kind of data is illustrated in Figure 2.1.

We here model the joint distribution $(W_{i1}, \ldots, W_{in_i})$ through the specification of the conditional laws $\mathcal{L}(W_{ij}|\boldsymbol{x}_{ij}, W_{i1}, \ldots, W_{ij-1})$.

2.2.1 Renewal Processes

Renewal processes are the canonical models for waiting times and are defined as processes for which the gap times of each patient W_{ij}, $j = 1, \ldots, n_i$ are i.i.d, conditionally to covariates and parameters. In other words, the intensity function is equal to the hazard rate, i.e. $\lambda(t|\mathcal{H}(t)) = h(t - T_{N(t^-)})$. This strong assumption corresponds to the setting in which individuals are restored to the original physical state after each event. This makes no sense in our investigation, which aims at discovering the influence of past events on patients. However, by extending renewal processes in various ways one can obtain other flexible models.

The likelihood function from n independent individuals is of the form

$$L = \prod_{i=1}^{n} \left[\prod_{j=1}^{n_i} f(w_{ij}|\boldsymbol{x}_{ij}) \right] S(w_{i,n_i+1}|\boldsymbol{x}_{i,n_i+1}).$$

2.2.2 Extensions and generalisations

In a more general case, when the assumption of independent gap times is unrealistic, models can be formulated through the sequence of conditional laws $\mathcal{L}(W_{ij}|\boldsymbol{x}_{ij}, W_{i1}, \ldots, W_{ij-1})$, $j = 1, 2, \ldots, n_i$. In this case, the cumulative distribution functions

$$F_j(w|\boldsymbol{x}_{ij}, w_i^{(j-1)}) = \mathbb{P}(W_{ij} \leq w|\boldsymbol{x}_{ij}, w_i^{(j-1)})$$

where $w_i^{(j-1)} = (w_{i1}, \ldots, w_{ij-1})'$, can change at each gap time. This format allows various types of dependence on previous event history to be considered, including elapsed time $w_{i1} + \cdots + w_{ij-1}$ up to $(j-1)^{st}$ event.

The two dominant families of models in this framework are AFT and PH regressions. For parametric models, the likelihood function from a set of n independent processes is

$$L = \prod_{i=1}^{n} \left[\prod_{j=1}^{n_i} f_j(w_{ij}|\boldsymbol{z}_{ij}) \right] S_{n_i+1}(w_{i,n_i+1}|\boldsymbol{z}_{i,n_i+1}),$$

where $\boldsymbol{z}_{ij}$ is a vector that models the dependence of W_{ij} on $\boldsymbol{x}_{ij}$ and $w_i^{(j-1)}$. Studies of this kind have already been proposed in a classical framework. Prentice et al. (1981) propose a PH model for recurrent events. This semiparametric model is obtained by specifying the intensity function as one of the following

$$\lambda(t|N(t), X(t)) = \lambda_{0s}(t) \exp(\boldsymbol{x}(t)\boldsymbol{\beta}_s)$$
$$\lambda(t|N(t), X(t)) = \lambda_{0s}(t - t_{n(t)}) \exp(\boldsymbol{x}(t)\boldsymbol{\beta}_s)$$

where $t_{n(t)}$ is the time of the preceding event. These two choices correspond to the natural time scales for the baseline hazard function: one is the time t from the beginning of the study and the other is $t - t_{n(t)}$, the time elapsed since the immediately preceding event. Moreover, the index s allows the baseline hazard to be stratum-specific. This is a more general case, as if one chooses $\lambda_{0s}(\cdot) = \lambda_0(\cdot)\ \forall s = 1, \ldots, J$ the case of a simple renewal process is obtained.

Chang and Wang (1999) propose a slightly different model by incorporating two kinds of covariates: some structural covariates (fixed) and some episode-specific covariates. For example, in a study of schizophrenia, gender and marital status may have the same effect for different episodes, but the age of disease onset may have distinct effects over different episodes. Moreover, the authors propose a strategy to maximise the partial likelihood.

The aim of this work, however, is to model distributions in the most flexible way. Therefore, the natural framework is Bayesian nonparametrics, which allows us to specify a non-parametric form for the laws of the gap times between recurrent events.

Chapter 3

A BNP model for recurrent events

In this chapter, the main goal is to present the Bayesian semiparametric model used in this work to represent gap times between recurrent events. After describing the model, the calculation of the full conditionals and the sampling scheme are illustrated. The corresponding MCMC algorithm to compute posterior inference will be tested on two simulated datasets in order to assess its functioning.

3.1 The model

Recalling the notation of the previous chapter, based on Cook and Lawless (2007), let W_{ij}, $j = 1, \dots, n_i$ denote the gap times between the $(j-1)^{th}$ and j^{th} event of patient i, $i = 1, \dots, n$. Each individual is observed over the time interval $[0, \tau_i]$, and $t = 0$ corresponds at the first event. By $\boldsymbol{x}_{ij}$ we denote a vector of p covariates for patient i at time j. If $n_i + 1$ events are observed at times $0 =: t_{i0} < t_{i1} < \cdots < t_{in_i} \leq \tau_i$, let $w_{ij} = t_{ij} - t_{ij-1}$ for $j = 1, \dots, n_i$ and $w_{i,n_i+1} = \tau_i - t_{in_i}$. These are the observed gap times for individual i with the final time being possibly censored. Let J denote the maximum number of observed repeated events, i.e. $J = \max_{i=1,\dots,n} n_i$.

First of all, let us transform the data with a log-function: $Y_{ij} = \log(W_{ij})$, $i = 1, \dots, n$, $j = 1, \dots, n_i$. In this model we describe the joint distribution $(Y_{i1}, \dots, Y_{in_i})$ through the specification of the conditional laws $\mathcal{L}(Y_{ij}|\boldsymbol{x}_{ij}, Y_{i1}, \dots, Y_{ij-1})$. In particular, we assume a dependence structure similar to an AR(1) model (cfr. Di Lucca et al., 2013). However, in this work the random intercepts and the coefficient parameters are free to vary for

each gap time. The model can be written as

$$\begin{aligned}
Y_{i1} &= \boldsymbol{x}_{i1}^T\boldsymbol{\beta}_1 + \alpha_{i12} + \sigma\epsilon_{i1} \\
Y_{i2} &= \boldsymbol{x}_{i2}^T\boldsymbol{\beta}_2 + \alpha_{i21}Y_{i1} + \alpha_{i22} + \sigma\epsilon_{i2} \\
Y_{i3} &= \boldsymbol{x}_{i3}^T\boldsymbol{\beta}_3 + \alpha_{i31}Y_{i2} + \alpha_{i32} + \sigma\epsilon_{i3} \\
&\dots \\
Y_{in_i} &= \boldsymbol{x}_{in_i}^T\boldsymbol{\beta}_{n_i} + \alpha_{in_i1}Y_{in_i-1} + \alpha_{in_i2} + \sigma\epsilon_{in_i}
\end{aligned} \tag{3.1}$$

where

$$\epsilon_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0,1).$$

Let us remark that each $\boldsymbol{\beta}_j$, as well as $\boldsymbol{x}_{ij}$ $\forall j = 1,\dots,J, i = 1,\dots,n$ are vectors of length p (the number of available covariates), and that $\boldsymbol{\alpha}_i = (\alpha_{i11}, \alpha_{i21}, \alpha_{i22}, \dots, \alpha_{iJ1}, \alpha_{iJ2})$ $\forall i = 1,\dots,n$ is a vector with length $2J-1$.

Conditionally to the parameter vector $(\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_J,\boldsymbol{\alpha}_i,\sigma^2)$, we assume that the observations $\boldsymbol{Y}_i = (Y_{i1},\dots,Y_{in_i})$ are independent. Therefore, this model is equivalent to

$$\begin{aligned}
&Y_{i1}|\boldsymbol{x}_{i1},\boldsymbol{\beta}_1,\boldsymbol{\alpha}_i,\sigma^2 \sim \mathcal{N}(\boldsymbol{x}_{i1}^T\boldsymbol{\beta}_1 + \alpha_{i12},\sigma^2) \\
&Y_{ij}|Y_{ij-1},\boldsymbol{x}_{ij},\boldsymbol{\beta}_j,\boldsymbol{\alpha}_i,\sigma^2 \sim \mathcal{N}(\boldsymbol{x}_{ij}^T\boldsymbol{\beta}_j + \alpha_{ij1}Y_{ij-1} + \alpha_{ij2},\sigma^2) \qquad \forall j = 2,\dots,n_i.
\end{aligned}$$

We now discuss the specification of the $\boldsymbol{\alpha}_i$ parameters. In order to model flexible distributions for the gap times, we assume that these parameters are a sample from a Dirichlet process, i.e.

$$\begin{aligned}
&\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_n|G \overset{\text{iid}}{\sim} G \\
&G \sim \text{DP}(MG_0)
\end{aligned}$$

In such case, the model can be rewritten as a DPM model:

$$\begin{aligned}
&\boldsymbol{Y}_i|\boldsymbol{\alpha}_i,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_J,\boldsymbol{x}_i,\sigma^2 \overset{\text{ind}}{\sim} k(\boldsymbol{y};\boldsymbol{\alpha}_i,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_J,\boldsymbol{x}_i,\sigma^2) = \mathcal{N}_J(\boldsymbol{\mu}_i,\Sigma_i) \\
&\sigma^2 \sim \text{inv-gamma}\left(\frac{\nu_0}{2},\frac{\nu_0\sigma_0^2}{2}\right) \\
&\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_J \overset{\text{iid}}{\sim} \mathcal{N}_p(\boldsymbol{0},\beta_0^2\mathbb{I}_p) \\
&\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_n|G \overset{\text{iid}}{\sim} G \\
&G \sim \text{DP}(MG_0)
\end{aligned} \tag{3.2}$$

This model implies that the data are distributed according to a mixture of kernels $k(\boldsymbol{y};\boldsymbol{\alpha}_i,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_J,\boldsymbol{x}_i,\sigma^2)$ where the mixing probability integrates with respect to the $\boldsymbol{\alpha}$ parameters. In the most common case (see Müller et al., 1996, for details) the mean and the covariance matrix of the kernel are directly sampled from the Dirichlet process.

In this case, however, the dependency of the moments of $k(\cdot)$ on the DP sample is more complicated.

One can easily prove (see Appendix A) that, given the model specified in (3.1), the mean vector and the covariance matrix of each gap time, given the parameters, are respectively

$$\boldsymbol{\mu}_i = \begin{pmatrix} \mathbb{E}[Y_{i1}|\boldsymbol{x}_{i1}, \boldsymbol{\beta}_1, \boldsymbol{\alpha}_i, \sigma] \\ \mathbb{E}[Y_{i2}|\boldsymbol{x}_{i2}, \boldsymbol{\beta}_2, \boldsymbol{\alpha}_i, \sigma] \\ \vdots \\ \mathbb{E}[Y_{iJ}|\boldsymbol{x}_{iJ}, \boldsymbol{\beta}_J, \boldsymbol{\alpha}_i, \sigma] \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{i1}^T\boldsymbol{\beta}_1 + \alpha_{i12} \\ \boldsymbol{x}_{i2}^T\boldsymbol{\beta}_2 + \alpha_{i21}(\boldsymbol{x}_{i1}^T\boldsymbol{\beta}_1 + \alpha_{i12}) + \alpha_{i22} \\ \vdots \\ \boldsymbol{x}_{iJ}^T\boldsymbol{\beta}_J + \alpha_{iJ1}\mathbb{E}[Y_{iJ-1}|\dots] + \alpha_{iJ2} \end{pmatrix},$$

and

$$\begin{aligned} \Sigma_i &= \begin{pmatrix} \mathrm{Var}[Y_{i1}|\dots] & \mathrm{Cov}(Y_{i1}, Y_{i2}) & \mathrm{Cov}(Y_{i1}, Y_{i3}) & \dots \\ & \mathrm{Var}[Y_{i2}|\dots] & \mathrm{Cov}(Y_{i2}, Y_{i3}) & \dots \\ & & \mathrm{Var}[Y_{i3}|\dots] & \\ & & & \ddots \end{pmatrix} \\ &= \sigma^2 \begin{pmatrix} 1 & \alpha_{i21} & \alpha_{i31}\alpha_{i21} & \dots \\ & 1 + \alpha_{i21}^2 & \alpha_{i31}(1 + \alpha_{i21}^2) & \dots \\ & & 1 + \alpha_{i31}^2 + \alpha_{i31}^2\alpha_{i21}^2 & \\ & & & \ddots \end{pmatrix}. \end{aligned}$$

Moreover, some preliminary comments are also necessary:

- The double indexing of $\boldsymbol{x}_{ij}$ denotes an implicit time-dependence of the covariates $\boldsymbol{x}_{ij}(t)$.
- Even if the covariates are fixed, we here allow for the covariate parameters $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_J$ to change for each gap time. In other words, we assume for a stratum-specific effect of the covariates.
- DP denotes a multivariate DP for the entire vector $\boldsymbol{\alpha}_i$. The ties in the $\boldsymbol{\alpha}_i$'s will induce a clustering of observations according to their entire trajectories.
- There are missing gap times for some individuals. Even if one observes only n_i gap times, the entire trajectories of length J have to be imputed, coherently with the Bayesian framework.

The base measure of the DP prior of the model (3.2) is

$$\begin{aligned} &G_0 = W_1 \otimes Z_2 \otimes W_2 \otimes \dots \otimes Z_J \otimes W_J, \\ &W_1, \dots W_J \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_W^2), \\ &Z_2, \dots Z_J \overset{\text{iid}}{\sim} \mathrm{Uniform}(a_Z, b_Z), \end{aligned} \tag{3.3}$$

where the components of the centering measure G_0 are the two families of independent r.v. $\{W_i\}_{i\geq 1} \perp \{Z_j\}_{j\geq 2}$, and σ_W^2 is a large variance parameter. We denote with W_i the components relative to the random intercepts $(\alpha_{12}, \alpha_{22}, \dots, \alpha_{J2})$ and with Z_i the components relative to the terms multiplying the previous gap times $(\alpha_{21}, \alpha_{31}, \dots, \alpha_{J1})$. Let us point out that we will choose $a_Z = -1$ and $b_Z = 1$, so that the support of the prior distribution on these latter parameters is $(-1, 1)$. Otherwise, the process would be non-stationary because its variance would asymptotically approach infinity.

The total number of parameters is $J(2+p)$. In fact, the dimension of $\boldsymbol{\alpha}$ is $2J-1$; each of the J covariate parameters $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_J$ has dimension p (the number of covariates); the last parameter of interest is the variance of the process σ^2.

It is well-known that non-parametric models suffer from the high sensitivity with respect to the choice of the base measure G_0. Therefore, it is advisable to carry out a robustness analysis for the prior specification. This issue will be tackled in Section 4.5.

3.2 Computational strategy

In this section, the MCMC strategy used to sample from the posterior distribution of the parameters is illustrated. We refer in the following to the model (3.2), together with the prior distributions (3.3).

3.2.1 Handling the non-conjugacy

As already discussed in Section 1.3.2, the aim of this work is to determine a Polya scheme in order to fit the model with the DP prior. The main ideas are based on Algorithm 8 in Neal (2000) and its adaptation in Argiento et al. (2009).

Since the dependence of the kernels $k(\boldsymbol{y}; \boldsymbol{\alpha}, \dots)$ on the $\boldsymbol{\alpha}$ parameters is complex, there does not exist a conjugate prior with respect to the base measure G_0. Therefore, when looking for a sampling strategy, one has to tackle the issue of calculating the marginal of the data

$$q_0(\boldsymbol{y}) = \int k(\boldsymbol{y}; \boldsymbol{\alpha}) G_0(d\boldsymbol{\alpha}) \tag{3.4}$$

in the label update step

$$\mathbb{P}(s_i' = j | \boldsymbol{s}_{-i}, \boldsymbol{y}_i, \boldsymbol{\alpha}_i, \dots) = \begin{cases} \dfrac{k(\boldsymbol{y}_i; \boldsymbol{\alpha}_j^*) d_j^{(-i)}}{M q_0(\boldsymbol{y}_i) + \sum_{h \in \boldsymbol{s}_{-i}} k(\boldsymbol{y}_i; \boldsymbol{\alpha}_h^*) d_h^{(-i)}} & \text{for } j \in \boldsymbol{s}_{-i} \\ \dfrac{M q_0(\boldsymbol{y}_i)}{M q_0(\boldsymbol{y}_i) + \sum_{h \in \boldsymbol{s}_{-i}} k(\boldsymbol{y}_i; \boldsymbol{\alpha}_h^*) d_h^{(-i)}} & \text{for } j = k_{new}, \end{cases} \tag{3.5}$$

where $d_j^{(-i)}$ denotes the size of the j^{th} cluster, discarding the i^{th} observation.

For these reasons, another strategy has been implemented. Using the same idea of Algorithm 8 in Neal (2000), which has been presented in Section 1.3.2, we approximate the marginal distribution $q_0(\cdot)$ using a Monte Carlo strategy. Accordingly, auxiliary variables are introduced in the MCMC scheme in order to evaluate via Monte Carlo the integral (3.4). The data augmentation consists in substituting, for each observation, the base measure G_0 with the auxiliary variables $\phi^{(1)}, \dots, \phi^{(n)}$, where each $\phi^{(i)} = (\boldsymbol{\phi}_1^{(i)}, \boldsymbol{\phi}_2^{(i)}, \dots, \boldsymbol{\phi}_m^{(i)})$ is a sample from G_0 of size m ($\phi^{(i)}$ is a $m \times (2J-1)$ matrix).

The choice of m has been discussed in Neal (2000): when $m = 1$, the algorithm closely resembles the "no-gaps" algorithm of MacEachern and Müller (1998); when $m \to \infty$ the Monte Carlo approximation is more precise. However, the equilibrium distribution of the Markov chain is correct for any value of m. In this work we use $m = 3$, which represents a good approximation at a feasible computational time.

3.2.2 Full conditionals

We illustrate here a general scheme of the Gibbs Sampler that allows us to sample from the posterior distribution. The details of each full conditional will be given in the subsections below.

Algorithm 1 Gibbs Sampler

function GIBBS($\mathbb{Y}, \mathbb{X}, \boldsymbol{n}, N, burnin, thin, a, m = 3$)
  Initialise the labels $\boldsymbol{s}$ with a k-means algorithm ($K = 10$)
  Initialise the parameters $\boldsymbol{\alpha}_j^*$ $\forall j = 1, \dots, k$ randomly sampling from $\mathcal{N}(0, 10)$
  Initialise the variance σ randomly sampling from $\mathcal{U}(0, 10)$
  Initialise the parameters $\boldsymbol{\beta}_j$ $\forall j = 1, \dots, J$ randomly sampling from $\mathcal{U}(-10, 10)$
  for $iter = 2$ **to** N **do**
    Neal's Algorithm step: $\boldsymbol{s}^{new} \leftarrow$ SAMP_CONF$(\boldsymbol{s}, rest)$ using (3.6)
    Shuffle step: $\boldsymbol{\alpha}_i^{new} \leftarrow$ SHUFFLE$(\boldsymbol{\alpha}_i, rest)$ using (3.7)
    Update clusters: $k^{new} \leftarrow$ UNIQUES$(\boldsymbol{\alpha}_i^{new})$
    Block-update step: $\boldsymbol{\alpha}_j^{*new} \leftarrow$ BLOCK_UPD$(\boldsymbol{\alpha}_j^*, rest)$ using (3.8) and (3.9)
    Gibbs step: $\sigma^{2new} \leftarrow$ SIGMA_UPD$(\sigma^2, rest)$ using (3.10)
    Gibbs step: $\boldsymbol{\beta}_j^{new} \leftarrow$ BETA_UPD$(\boldsymbol{\beta}_j, rest)$ using (3.11)
  Discard the burnin and thin the chain
  return $s, k, \alpha^2, \sigma, \beta$

Update labels

The labels are updated using two full conditionals. The first one is

$$\mathbb{P}(s_i^{'} = j|\boldsymbol{s}_{-i}, \boldsymbol{\alpha}_i, \dots) \propto \begin{cases} k(\boldsymbol{y}_i; \boldsymbol{\alpha}_j^*) d_j^{(-i)} & \text{for } j \in \boldsymbol{s}_{-i}, \\ \\ \frac{M}{m} \sum_{h=1}^{m} k(\boldsymbol{y}_i; \boldsymbol{\phi}_h^{(i)}) & \text{for } j = k_{new}. \end{cases} \tag{3.6}$$

We remark that this expression is similar to (3.5), with the only difference that here the integral is approximated via Monte Carlo. Equation (3.6) implies that, for each observation, the new label is, alternatively:

- sampled from one of the existing "old" labels in $\boldsymbol{s}_{-i}$. In this case, if the observation was in a single cluster, the number of clusters decreases by one;
- a new value. In this case, unless the observation was in a single cluster, the number of clusters increases by one.

The second full conditional is

$$\mathbb{P}(d\boldsymbol{\alpha}_i^{'}|\boldsymbol{s}^{'}, \boldsymbol{\alpha}_{-i}, \phi^{(1)}, \dots, \phi^{(n)}, \dots) \propto \begin{cases} \delta_{\boldsymbol{\alpha}^*_{s_i^{'}}}(d\boldsymbol{\alpha}_i^{'}) & \text{if } s_i^{'} \in \boldsymbol{s}_{-i} \\ \\ \sum_{h=1}^{m} k(\boldsymbol{y}_i; \boldsymbol{\phi}_h^{(i)}) \delta_{\boldsymbol{\phi}_h^{(i)}}(d\boldsymbol{\alpha}_i^{'}) & \text{if } s_i^{'} = k_{new} \end{cases} \tag{3.7}$$

Equation (3.7), instead, is needed in order to exchange the cluster specific parameters, once the new labels have been sampled. In particular:

- if the new label is sampled from one of the existing "old" labels in $\boldsymbol{s}_{-i}$, say j, the corresponding parameter is $\boldsymbol{\alpha}_j^*$;
- if the new label is a new value, the new parameter is equal to the corresponding augmented parameter $\boldsymbol{\phi}_h^{(i)}$ (sampled from G_0).

Update $\boldsymbol{\alpha}^*$

Once the new labels have been updated and the parameters have been exchanged, it is necessary to sample the unique cluster specific parameters $\boldsymbol{\alpha}_j^*$ from the proper full conditional using all the observations attributed to the h^{th} cluster, with

$$\mathcal{L}(\boldsymbol{\alpha}_h^*|\boldsymbol{s}, \dots) \propto G_0(\boldsymbol{\alpha}_h^*) \prod_{i:s_i=h} k(\boldsymbol{y}_i; \boldsymbol{\alpha}_h^*) \quad \forall h = 1, \dots, k.$$

For this purpose, since the model does not admit a conjugate prior, it is necessary to use a Metropolis step within the Gibbs sampler.

In light of the number of choices to be done when using a Metropolis-Hastings algorithm (e.g. the choice of the proposal), we propose a convenient strategy in order to update the parameters. Instead of performing a simultaneous update of the vectorial quantities $\boldsymbol{\alpha}_h^*$, one can update each component of the vector one at the time. Thus, the proposal densities have to be chosen in one dimensional spaces and not in $2J-1$-dimensional spaces. Moreover, for the components of $\boldsymbol{\alpha}^*$ relative to the intercept terms, a conjugate form can be obtained.

Therefore, the full conditionals for the components of $\boldsymbol{\alpha}^*$ with support in $(-1,1)$ become, $\forall h = 1,\dots,k$

$$\begin{aligned}
&\mathcal{L}(d\alpha_{h,2}^{*new}|\boldsymbol{s},\dots) \propto g_{0,2}(\alpha_{h,2}^{*new}) \prod_{i:s_i=h} k(\boldsymbol{y}_i;\boldsymbol{\alpha}_h^{*new}) \\
&\vdots \\
&\mathcal{L}(d\alpha_{h,2J-2}^{*new}|\boldsymbol{s},\dots) \propto g_{0,2J-2}(\alpha_{h,2J-2}^{*new}) \prod_{i:s_i=h} k(\boldsymbol{y}_i;\boldsymbol{\alpha}_h^{*new}),
\end{aligned} \tag{3.8}$$

where at each step the new value $\boldsymbol{\alpha}_h^{*new}$ differs from the old one only at the component $j \in \{2,4,\dots,2J-2\}$ considered. For each cluster h, and for each component j of the vector $\boldsymbol{\alpha}_h^*$, the proposal is a univariate normal distribution $\mathcal{N}(\alpha_{hj}^*, [\Sigma_p]_{j,j})$, where Σ_p is computed as minus the inverse of the Hessian of the log-posterior around its maximum. Mathematically, $\Sigma_p = -H^{-1}$, where

$$H_{ij} = \frac{\partial \log(f(\boldsymbol{y};\boldsymbol{\alpha}))}{\partial\alpha_i \partial\alpha_j}.$$

Moreover, in order to enhance the convergence of the Metropolis-within-Gibbs steps, the normal distributions have been truncated on the domain interval $(-1,1)$. This leads to a faster algorithm because it avoids the proposal of values that are successively refused with probability 1.

As far as the other components are concerned, $\forall j = 1,3,\dots,2J-1$, $\forall h = 1,\dots,k$ the framework is the following

$$\begin{aligned}
&Y_{ij}|Y_{ij-1},\boldsymbol{x}_{ij},\boldsymbol{\beta}_j,\boldsymbol{\alpha}_i,\sigma^2 \overset{\text{ind}}{\sim} \mathcal{N}(\boldsymbol{x}_{ij}^T\boldsymbol{\beta}_j + \alpha_{ij1}Y_{ij-1} + \alpha_{ij2}, \sigma^2) \quad \forall i \text{ s.t. } s_i = h \\
&\alpha_{ij2} \sim \mathcal{N}(0,\sigma_W^2),
\end{aligned}$$

and therefore the posterior is

$$\mathcal{L}(\alpha_{ij2}|rest) = \mathcal{N}\left(\frac{\sigma_W^2 \sum_{i:s_i=h} \hat{y}_{ij}}{\sigma_W^2 d_h + \sigma^2}, \frac{\sigma_W^2\sigma^2}{\sigma_W^2 d_h + \sigma^2}\right), \tag{3.9}$$

where the $\hat{y}_i$'s are the scaled data, i.e.

$$\begin{aligned}
\widehat{Y}_{i1} &= Y_{i1} - \boldsymbol{x}_{i1}^T\boldsymbol{\beta}_1 \\
\widehat{Y}_{ij} &= Y_{ij} - \boldsymbol{x}_{ij}^T\boldsymbol{\beta}_j - \alpha_{ij1}\hat{Y}_{ij-1}.
\end{aligned}$$

Update σ^2

As far as the update of the variance parameter is concerned, the choice of an appropriate prior distribution allows us to find a conjugate full conditional. In fact, given the model

$$\begin{aligned}
&\boldsymbol{Y}_i|\boldsymbol{\alpha}_i, \boldsymbol{x}_i, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J, \sigma^2 \overset{\text{ind}}{\sim} \mathcal{N}_J(\boldsymbol{\mu}_i, \Sigma_i) \\
&\sigma^2 \sim \text{inv-gamma}\left(\frac{\nu_0}{2}, \frac{\nu_0\sigma_0^2}{2}\right),
\end{aligned}$$

it is straightforward (see Appendix A) to prove that

$$\mathcal{L}(\sigma^2|rest) = \text{inv-gamma}\left(\frac{\nu_0 + nJ}{2}, \frac{\nu_0\sigma_0^2 + \sum_{i=1}^{n}(\boldsymbol{y}_i - \boldsymbol{\mu}_i)^T(\boldsymbol{y}_i - \boldsymbol{\mu}_i)}{2}\right). \tag{3.10}$$

Update $\boldsymbol{\beta}$

The part of the model involved in the update of the covariate parameters is

$$\begin{aligned}
&\boldsymbol{Y}_i|\boldsymbol{\alpha}_i, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J, \boldsymbol{x}_i, \sigma^2 \overset{\text{ind}}{\sim} \mathcal{N}_J(\boldsymbol{\mu}_i, \Sigma_i) \\
&\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J \overset{\text{iid}}{\sim} \mathcal{N}_p(\mathbf{0}, \beta_0^2 \cdot \mathbb{I}_p).
\end{aligned}$$

For each observation, let us define the transformed data

$$\begin{aligned}
&\widetilde{Y}_{i1} = Y_{i1} - \alpha_{i11} \\
&\widetilde{Y}_{i2} = Y_{i2} - \alpha_{i21}Y_{i1} - \alpha_{i22} \\
&\vdots \\
&\widetilde{Y}_{iJ} = Y_{iJ} - \alpha_{iJ1}Y_{iJ-1} - \alpha_{iJ2}
\end{aligned}$$

and the vectorial quantities

$$\begin{aligned}
&\widetilde{\boldsymbol{Y}}_1 = (\widetilde{Y}_{11}, \widetilde{Y}_{21}, \ldots, \widetilde{Y}_{n1}) \\
&\vdots \\
&\widetilde{\boldsymbol{Y}}_j = (\widetilde{Y}_{1j}, \widetilde{Y}_{2j}, \ldots, \widetilde{Y}_{nj}) \quad \forall j = 2, \ldots, J.
\end{aligned}$$

Hence we have

$$\begin{aligned}
&\widetilde{\boldsymbol{Y}}_1|\mathbb{X}, \boldsymbol{\beta}_1, \sigma^2, \boldsymbol{\alpha} \sim \mathcal{N}_n(\mathbb{X}\boldsymbol{\beta}_1, \sigma^2\mathbb{I}_n) \\
&\widetilde{\boldsymbol{Y}}_j|\boldsymbol{Y}_{j-1}, \mathbb{X}, \boldsymbol{\beta}_j, \sigma^2, \boldsymbol{\alpha} \sim \mathcal{N}_n(\mathbb{X}\boldsymbol{\beta}_j, \sigma^2\mathbb{I}_n) \quad \forall j = 2, \ldots, J.
\end{aligned}$$

Thus, for each gap time, we are in the case of a classical univariate linear model and we can update each covariate parameter one at the time, i.e.

$$\begin{cases} \widetilde{\boldsymbol{Y}}_1|\mathbb{X},\boldsymbol{\beta}_1,\sigma^2,\boldsymbol{\alpha}\sim\mathcal{N}_n(\mathbb{X}\boldsymbol{\beta}_1,\sigma^2\mathbb{I}_n) \\ \boldsymbol{\beta}_1\sim\mathcal{N}_p(\mathbf{0},\beta_0^2\mathbb{I}_p) \end{cases} \Rightarrow \boldsymbol{\beta}_1|\widetilde{\boldsymbol{Y}}_1,rest\sim\mathcal{N}_p(\boldsymbol{b}_{1n},B_n),$$

$$\begin{cases} \widetilde{\boldsymbol{Y}}_j|\boldsymbol{Y}_{j-1},\mathbb{X},\boldsymbol{\beta}_j,\sigma^2,\boldsymbol{\alpha}\sim\mathcal{N}_n(\mathbb{X}\boldsymbol{\beta}_j,\sigma^2\mathbb{I}_n) \\ \boldsymbol{\beta}_j\sim\mathcal{N}_p(\mathbf{0},\beta_0^2\mathbb{I}_p) \end{cases} \Rightarrow \boldsymbol{\beta}_j|\widetilde{\boldsymbol{Y}}_j,rest\sim\mathcal{N}_p(\boldsymbol{b}_{jn},B_n)\quad\forall j=2,\dots,J, \tag{3.11}$$

where, as in the case of linear models with known variance,

$$B_n=\left(\frac{\mathbb{X}^T\mathbb{X}}{\sigma^2}+\frac{\mathbb{I}_p}{\beta_0^2}\right)^{-1},$$
$$\boldsymbol{b}_{in}=\left(\frac{\mathbb{X}^T\mathbb{X}}{\sigma^2}+\frac{\mathbb{I}_p}{\beta_0^2}\right)^{-1}\left(\frac{\mathbb{X}^T\widetilde{\boldsymbol{y}}_i}{\sigma^2}\right)=B_n\frac{\mathbb{X}^T\widetilde{\boldsymbol{y}}_i}{\sigma^2}.$$

3.2.3 Optimal partition

Once the MCMC chain approximating the posterior distribution is obtained, our main goal is to attribute each observation to a cluster, i.e. to find a cluster estimate. In this model we have already detailed a way to obtain a sample from the posterior distribution of the labels of the data $\boldsymbol{\rho}_n^{(1)},\dots,\boldsymbol{\rho}_n^{(N_{samp})}$, where $\boldsymbol{\rho}_n^{(i)}=(s_1^{(i)},\dots,s_n^{(i)})$. Nevertheless, since the support of $\boldsymbol{\rho}_n$ is a discrete space with large cardinality (the Bell number), the choice of a point estimate is an issue that should not be overlooked. The posterior mode, for example, is not an adequate solution as each support point might have a negligible posterior probability.

In literature, there exist many papers dealing with this problem. We refer, in particular, to Lau and Green (2007) as done in Argiento et al. (2014). This approach is the following: a suitable loss function $L(\boldsymbol{\rho}_n,\hat{\boldsymbol{\rho}}_n)$ is introduced, giving the cost of estimating the "true" $\boldsymbol{\rho}_n$ by $\hat{\boldsymbol{\rho}}_n$. Then, the proposed estimate is given by any partition $\hat{\boldsymbol{\rho}}_n$ which minimises the posterior expectation of the loss function, i.e.

$$\hat{\boldsymbol{\rho}}_n\in\arg\min_y\mathbb{E}[L(\boldsymbol{\rho}_n,y)|\text{data}].$$

We here use Binder's loss function (cfr. Binder, 1978, for details), assigning cost b when two observations are wrongly clustered together and cost a when two observations are erroneously assigned to different clusters, that is

$$L(\boldsymbol{\rho}_n,\hat{\boldsymbol{\rho}}_n)=\sum_{i<j\leq n}\left(a\mathcal{I}_{\{s_i=s_j;\widehat{s}_i\neq\widehat{s}_j\}}+b\mathcal{I}_{\{s_i\neq s_j;\widehat{s}_i=\widehat{s}_j\}}\right).$$

It is not difficult to see (Lau and Green, 2007) that, taking the expected value of both sides, one obtains

$$l(\hat{\boldsymbol{\rho}}_n) = \mathbb{E}[L(\boldsymbol{\rho}_n, \hat{\boldsymbol{\rho}}_n)|\text{data}] = a \sum_{i<j\leq n} p_{ij} - (a+b) \sum_{i<j\leq n} \mathcal{I}_{\{\widehat{s}_i=\widehat{s}_j\}}(p_{ij} - K), \tag{3.12}$$

where $\{p_{ij}\}$ is the unknown matrix of the posterior incidence probabilities $p_{ij} = \mathbb{P}(s_i = s_j|\text{data})$ and $K = b/(a+b) \in [0,1]$. Equation (3.12) can be rewritten as

$$l(\hat{\boldsymbol{\rho}}_n) = a \sum_{i<j\leq n} p_{ij} - (a+b)g(\hat{\boldsymbol{\rho}}_n), \tag{3.13}$$

highlighting the only term $g(\hat{\boldsymbol{\rho}}_n)$ that depends on the partition. Minimising $l(\hat{\boldsymbol{\rho}}_n)$ corresponds to maximising $g(\hat{\boldsymbol{\rho}}_n)$, with respect to $\hat{\boldsymbol{\rho}}_n$. However, the problem is that $\{p_{ij}\}$ is unknown.

We here propose a two-step method that allows us to estimate the matrix of the posterior incidence probabilities and then to choose the optimal partition:

1. half of the MCMC chain (which has a total length of N_{samp}) is used in order to estimate the probabilities $\widehat{p}_{ij} = \dfrac{\#\{s_i = s_j\}}{N_{samp}/2}$;
2. for every partition $\hat{\boldsymbol{\rho}}_n$ in the second half of the chain, we calculate the values $g(\hat{\boldsymbol{\rho}}_n) = \sum_{i<j\leq n} \mathcal{I}_{\{\widehat{s}_i=\widehat{s}_j\}}(\widehat{p}_{ij} - K)$; the optimal partition is the one that realises the maximum.

In the following, we choose $K = 0.5$, which corresponds to the assumption that the two misclassification costs are equal.

3.2.4 Implementation in the Julia language

We here justify the choice of the programming language used in this work. The R software is the classical tool when dealing with statistical analysis, as it provides user-submitted packages for specific functions or specific areas of study. However, Bayesian non-parametric models are often computationally infeasible for such a high-level language. R programming routines encourage operating on whole objects (i.e. vectorised code) because *while* and *for* loops are notoriously slow. Nevertheless, MCMC are not easily vectorised as every iteration depends on the previous one. Therefore, it would be advisable to carry out all the simulations in a lower-level programming language such as C or C++, or in any other language that has efficient loop structures.

Julia is a high-performance dynamic programming language for technical computing, with syntax that is familiar to users of other technical computing environments. The Julia language manages to combine computational efficiency with the easy scripting

and interpretation typical of any other high-level programming language. It provides a sophisticated compiler, distributed parallel execution, numerical accuracy, and an extensive mathematical function library (among the others, linear algebra and random number generator libraries are used in order to carry out a Bayesian analysis). For further details, see Bezanson et al. (2014).

Apart from its computational efficiency, another advantage of Julia over R is that objects are passed to functions by reference and not by copy. This issue, that may seem negligible at a first glance, is crucial in MCMC simulations which deal with many parameters, since high-dimensional matrices have to be handled at every iteration. Therefore, passing the memory address of such large objects instead of copying them in every function environment allows the algorithm to reduce the memory usage.

3.3 Simulated dataset 1

In order to check the validity of the model and algorithm proposed in the previous section, two different simulated datasets have been fitted to the model (3.2) - (3.3). In the first setting $n = 200$ data, with exactly $J = 3$ recurrent events, were generated from (3.1) each one with one of the three parameters $(\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \boldsymbol{\alpha}^3)$ with probability $(1/3, 1/3, 1/3)$. The complete parameter setting is:

$$\begin{aligned}
\boldsymbol{\alpha}^1 &= (\ \ 3.0 \quad\ 0.6 \quad\ 2.2 \quad\ 0.4 \quad\ 1.5\ \ ) \\
\boldsymbol{\alpha}^2 &= (-3.0 \quad -0.1 \quad -1.5 \quad -0.9 \quad -2.0\ \ ) \\
\boldsymbol{\alpha}^3 &= (\ \ 6.0 \quad -0.9 \quad\ 4.2 \quad\ 0 \quad\ 4.5\ \ ) \\
\sigma &= 1.5 \\
\boldsymbol{\beta}_1 &= \cdots = \boldsymbol{\beta}_J = \mathbf{0},
\end{aligned}$$

where we denoted with $\boldsymbol{\alpha}^k = (\alpha^k_{12}, \alpha^k_{21}, \alpha^k_{22}, \ldots, \alpha^k_{J1}, \alpha^k_{J2})$ the k^{th} possible value for $\boldsymbol{\alpha}$.

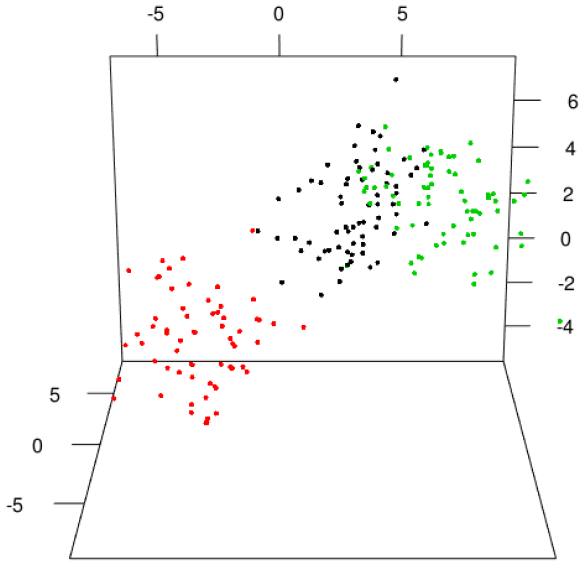


Figure 3.1: First simulated dataset in the space $\mathbb{R}^3$ of the 3 gap times.

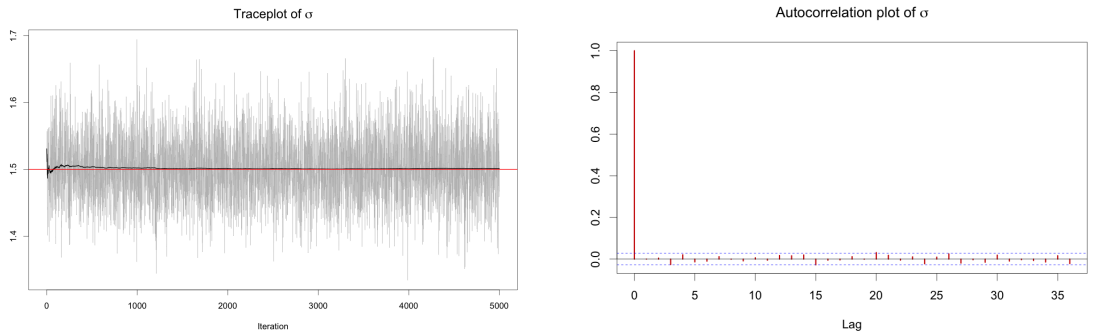
In this first case, no covariates are available and therefore this represents a simplification of the complete model presented in (3.2). Since the value of $J = 3$ was chosen, the data $\boldsymbol{y}_i$ can be represented in a three-dimensional space where each component is one of the three gap times (see Figure 3.1).

The hyperparameters are

$$\begin{aligned}
&\beta_0 = 10;\\
&\nu_0 = 2; \quad \sigma_0 = 1;\\
&M = 1;\\
&\sigma_W^2 = 100; \quad a_Z = -1; \quad b_Z = 1.
\end{aligned}$$

This choice is made in order to specify vague prior distributions. For example, the total mass parameter is $M = 1$ which, as illustrated in Figure 1.1, corresponds to a weak degree of belief in the base measure G_0. Furthermore, even the marginal components of G_0 are non-informative: the normal distributions W_i have large variances, and the Z_i's are uniform distributions over the interval $(-1, 1)$.

Posterior estimates are computed via the Gibbs sampler algorithm presented in Section 3.2.2. We run the algorithm in Julia for $70,000$ iterations, while the first $20,000$ iterations were discarded and we used a thinning of 10 to reduce the autocorrelation of the Markov chain. The final sample size is then 5000. Diagnostic convergence tests were done.

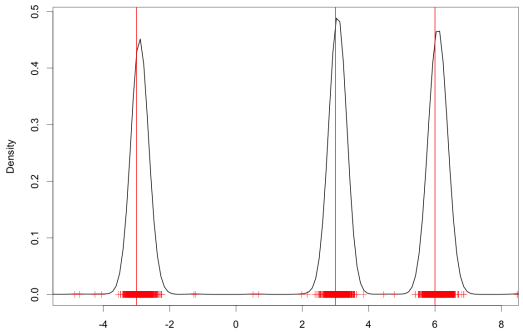


(a) Traceplot of σ. *(b) Autocorrelation function of σ.*

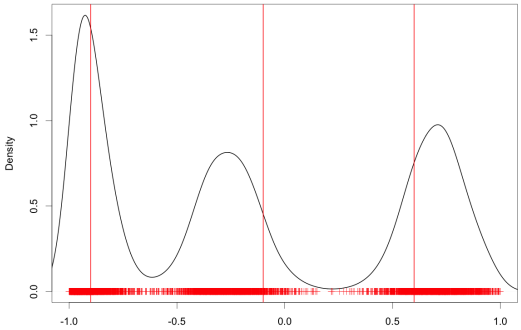
Figure 3.2: Output analysis used for a convergence check of the MCMC chain.

We report in Figure 3.2 traceplots and estimated autocorrelation functions of σ. As one can see, the chain seems to be stationary, as the traceplot is thick and the correlation between successive values of the sample is negligible.

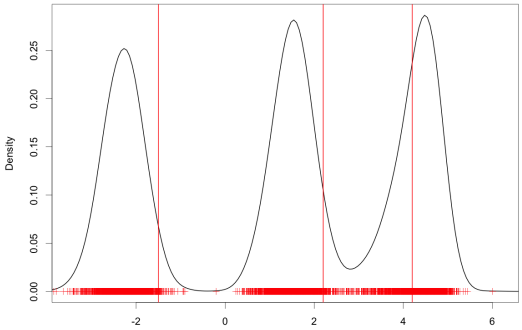
We do not show here the same output analysis concerning the $\boldsymbol{\alpha}$ parameters, as the traceplots are severely affected by the label switching problem (see Jasra et al., 2005, for further details). For this reason, inference on those parameters is displayed via the predictive distribution. Once the MCMC chain has been sampled, for each iteration one



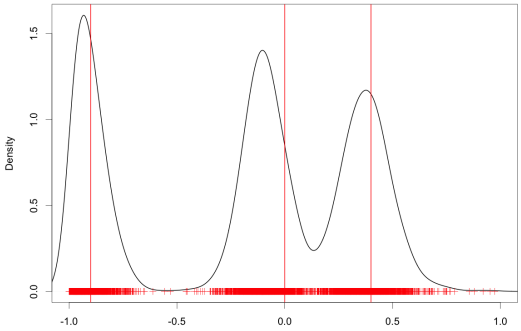
(a) Predictive density of α_{12}.



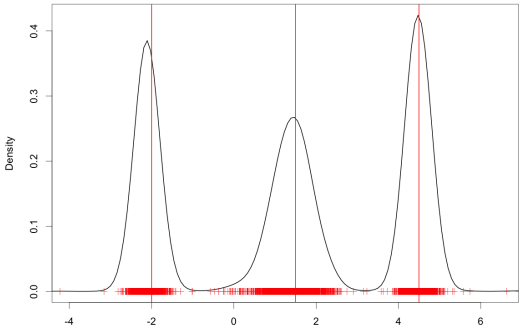
(b) Predictive density of α_{21}.



(c) Predictive density of α_{22}.



(d) Predictive density of α_{31}.



(e) Predictive density of α_{32}.

Figure 3.3: In black solid line, kernel density estimates of the predictive distributions of the $\boldsymbol{\alpha}$ parameters. In red vertical lines, the true values. The red ticks on the x-axis represent the sampled values.

value of $\boldsymbol{\alpha}$ is obtained via

$$\boldsymbol{\alpha}^{new(g)}|\boldsymbol{\alpha}_1^{(g)},\dots,\boldsymbol{\alpha}_n^{(g)} \sim \frac{M}{M+n}G_0 + \frac{\sum_{i=1}^{n}\delta_{\boldsymbol{\alpha}_i^{(g)}}}{M+n} \qquad \forall g = 1,\dots,N, \tag{3.14}$$

where $\boldsymbol{\alpha}^{new} = (\alpha_{12}^{new}, \alpha_{21}^{new}, \alpha_{22}^{new}, \dots, \alpha_{J1}^{new}, \alpha_{J2}^{new})$ and the superscript (g) denotes the current iteration of the chain.

The results of this procedure are displayed in Figure 3.3. The three components

of the mixture are clearly visible for each component of $\boldsymbol{\alpha}^{new}$. The sampled values of the predictive distributions are located around the true parameters. Furthermore, the predictive distributions are clearly different from the prior distribution of the parameters. In fact we supposed that, a priori, the components of $\boldsymbol{\alpha}$ relative to the intercept terms were univariate normal distributions centred in 0 and with variance equal to 100. As far as the components with support in $(-1, 1)$ are concerned, the uniform distribution that was used represents a non-informative prior.

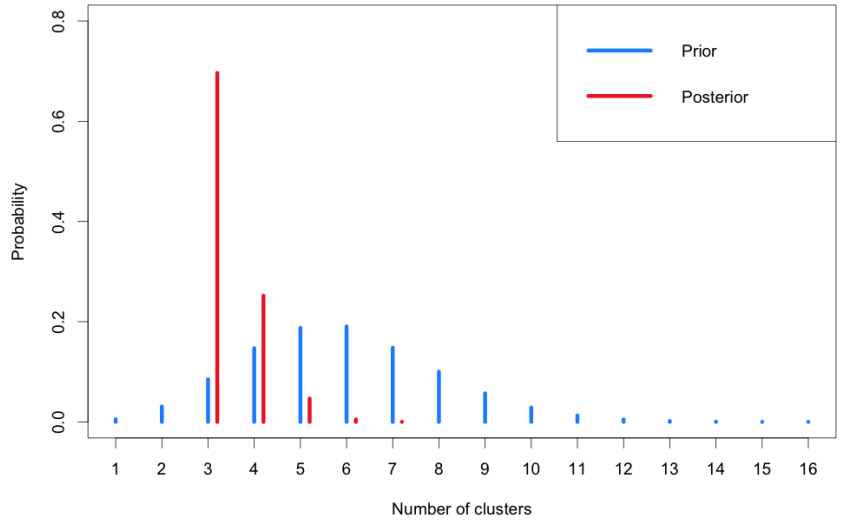


Figure 3.4: Prior and posterior number of clusters, i.e. of the number of unique values in the $\boldsymbol{\alpha}_i$'s.

In Figure 3.4, the posterior of the number of clusters is displayed. As one can see, the model provides a good estimate of the number of groups in the trajectories of the patients, that is 3. As it is documented in literature, this kind of models slightly overestimates the number of clusters. However, the posterior mode is located in 3 with a probability of around 0.7, which can be considered a very precise result.

The posterior distribution of σ, along with the 95% credible intervals shown in 3.5, is centred around the true value.

We now discuss the results of the method proposed in Section 3.2.3 that was used in order to estimate the optimal partition. In Figure 3.6 the six most recurrent partitions are displayed.

The first one, which is the posterior mode of the labels, has a posterior probability of 0.0044, which means that it appears 22 times out of the sample of size 5000. It is clear that this probability is too low to represent a good estimate for the optimal partition. For this reason, another approach has been used. Introducing Binder's loss function and

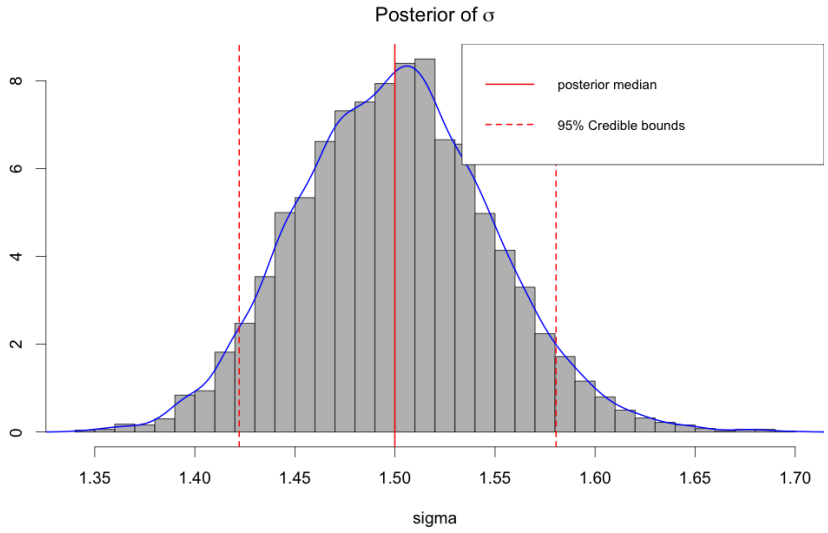


Figure 3.5: In blue solid line, the posterior distribution of the parameter σ, whose true value is 1.5. *A point estimate (the posterior median) and the* 95% *credible bounds are overlaid in red.*

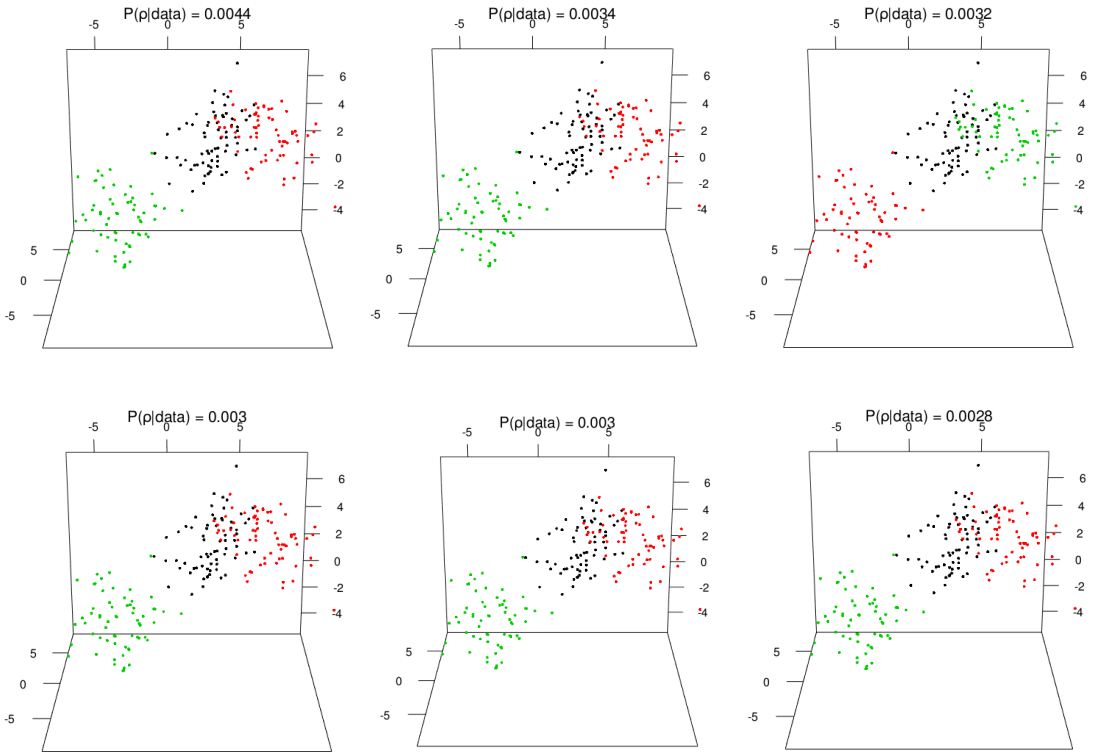
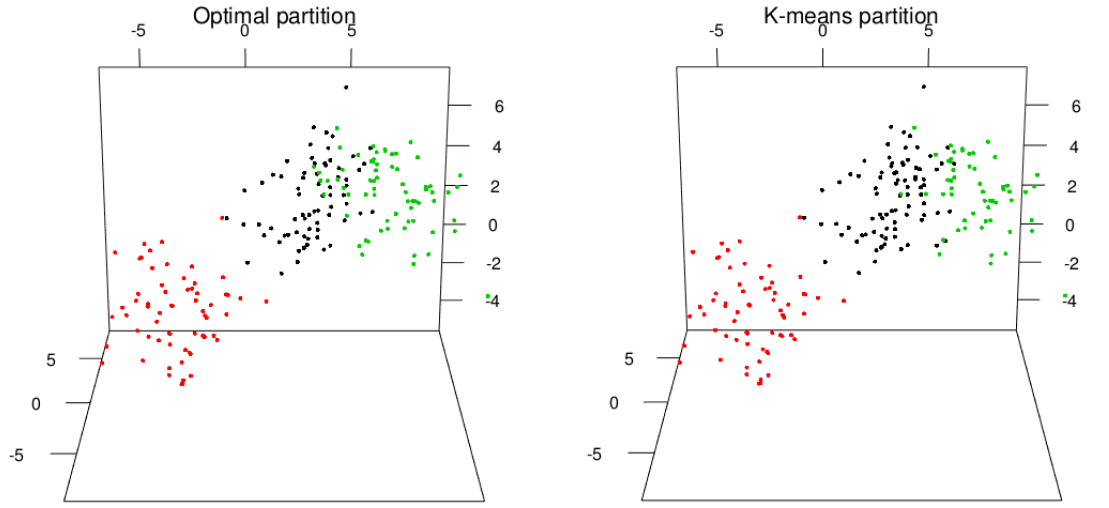


Figure 3.6: First six most recurrent partitions (most probable partitions), whose probabilities of occurrence are indicated above. The top-left figure is the posterior mode of the partitions $\boldsymbol{\rho}_n$.

evaluating it on the posterior sample, we estimated the optimal partition according to this criterion.

In Figure 3.7a and 3.7b, two partitions are compared: the first one is the Bayesian estimate as described above, while the other is obtained by the k-means algorithm. On the simulated dataset, as expected, our clustering method provides better results. Since the true partition is known, we can calculate the number of correctly classified data (i.e. the precision). Our method, with a precision of 98.0%, outperforms the k-means algorithm which classifies correctly 184 observations out of 200.



(a) Binder's loss function criterion. In this case, the percentage of correctly clustered data is 98.0%.

(b) K-means algorithm criterion. In this case, the percentage of correctly clustered data is 92.0%.

Figure 3.7: Comparision between the optimal partition according to Binder's loss function criterion and to a frequentist method.

Let us also remark that the optimal partition according to Binder's loss function does not necessarily correspond to the posterior mode of the parameters, nor to any of the partitions displayed in Figure 3.6. The reason is that the support of $\boldsymbol{\rho}_n$ is a discrete space with such a large cardinality that the state corresponding to the optimal partition could be rarely "reached" by the algorithm.

3.4 Simulated dataset 2

The second simulated dataset is more complex than the previous one. In fact, here covariates $\boldsymbol{x}_{ij}$ of length $p = 3$ are introduced in the model, and the number of gap times is allowed to be different for each observation. We generate $n = 200$ data from (3.1), each with n_i recurrent events and with one of the two parameters $(\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2)$ with probability $(1/2, 1/2)$. The complete parameter setting is:

$$
\begin{aligned}
\boldsymbol{\alpha}^1 &= (\ \ 3.0 \quad 0.6 \quad 2.2 \quad 0.3 \quad 1.5 \quad 0 \quad 3.0 \quad 0.95 \quad -1.0\ ) \\
\boldsymbol{\alpha}^2 &= (-3.0 \quad -0.1 \quad -1.5 \quad -0.2 \quad -2.0 \quad -0.95 \quad 0 \quad 0 \quad 2.0\ ) \\
\sigma &= 1.0 \\
\boldsymbol{\beta}_1 &= (-2.0 \quad -1.5 \quad -1.0\ ) \\
\boldsymbol{\beta}_2 &= (\ \ 0 \quad -0.2 \quad 0.2\ ) \\
\boldsymbol{\beta}_3 &= (\ \ 2.0 \quad 1.3 \quad 1.0\ ) \\
\boldsymbol{\beta}_4 &= (\ \ 1.0 \quad 0 \quad -1.0\ ) \\
\boldsymbol{\beta}_5 &= (\ \ 4.0 \quad -1.0 \quad -2.0\ )
\end{aligned}
$$

Let us remark here that the covariate parameters $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J$ are allowed to change for every gap time, even if the covariates are fixed. In other words, we are assuming that the effect of the same covariate can change over time. Moreover, the covariates are generated according to

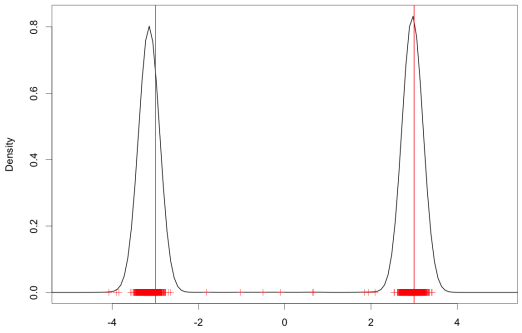
$$
\begin{aligned}
x_{ij,1} &\overset{\text{iid}}{\sim} \mathcal{N}(0, 2) \\
x_{ij,2} &\overset{\text{iid}}{\sim} Ber(0.5) \\
x_{ij,3} &\overset{\text{iid}}{\sim} Ber(0.2).
\end{aligned}
$$

We notice here that the second and the third covariate were sampled by two Bernoulli distributions with different parameters, in the attempt of simulating the behaviour of a categorical variable.

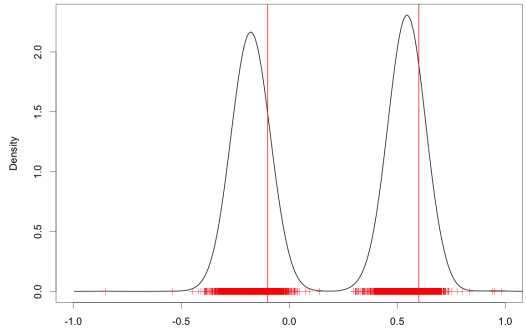
j	1	2	3	4	5
n_j	27	55	77	31	10

Table 3.1: Numer of observations with exactly j gap times, $j = 1, \ldots, J$.

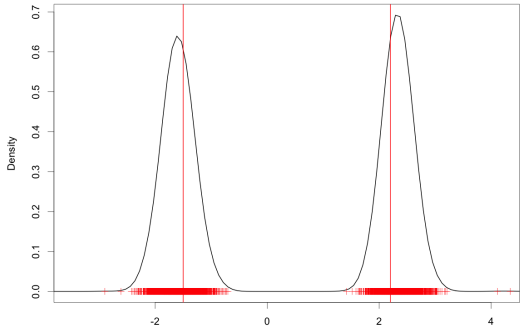
The number of gap times for each observation was sampled by a discrete distribution with values in $\{1, 2, 3, 4, 5\}$. In Table 3.1, the number of observations with at least j gap times is represented. We notice that most of the observations experience $j = 3$ recurrent events, as in the previous setting. However, since $J = \max_{i=1,\ldots,n} n_i = 5$, the number of



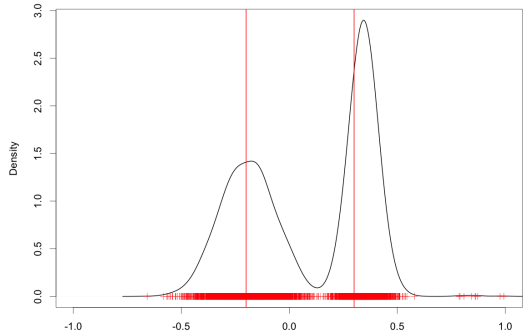
(a) Predictive density of α_{12}.



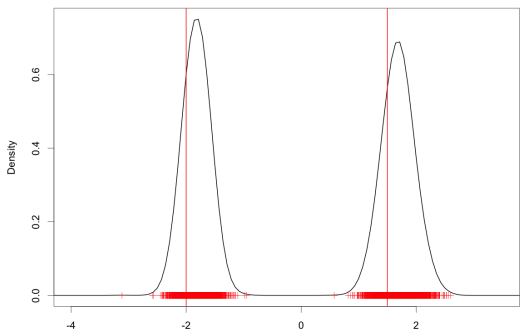
(b) Predictive density of α_{21}.



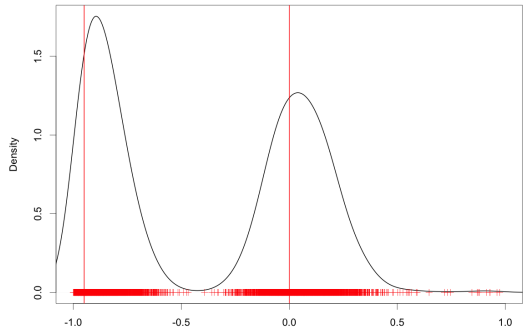
(c) Predictive density of α_{22}.



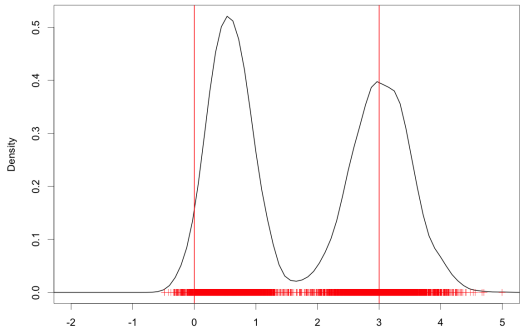
(d) Predictive density of α_{31}.



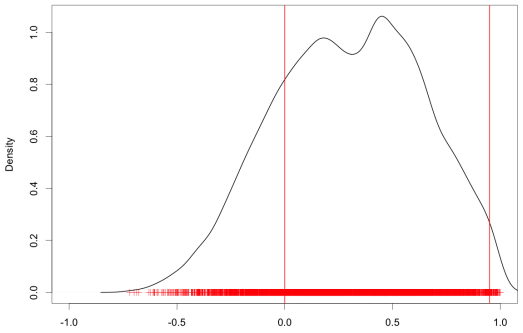
(e) Predictive density of α_{32}.



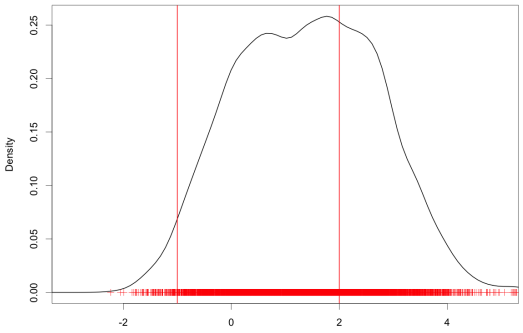
(f) Predictive density of α_{41}.



(g) Predictive density of α_{42}.



(h) Predictive density of α_{51}.



(i) Predictive density of α_{52}.

Figure 3.8: In black solid line, kernel density estimates of the predictive distributions of the $\boldsymbol{\alpha}$ parameters. In red vertical lines, the true values. The red ticks on the x-axis represent the sampled values.

cluster specific parameters to estimate is now $2J - 1 = 9$. In fact, missing data are imputed, coherently with the Bayesian framework, as described in Section 4.2.1.

Posterior estimates are computed in Julia via the Gibbs sampler algorithm presented in Section 3.2.2. We run the algorithm for $140,000$ iterations, while the first $40,000$ iterations were discarded and we used a thinning of 20 to reduce the autocorrelation of the Markov chain. The final sample size is then 5000. Some diagnostic convergence tests were done.

In Figure 3.8 the predictive distributions of the parameters, obtained via (3.14), are displayed. Let us do some remarks about the results. As one can expect, the precision of the first components of the parameter $\boldsymbol{\alpha}$ is greater than the precision of the last ones. In fact, for those first components all the observations are used in the Metropolis-within-Gibbs update. Instead, α_{51} and α_{52} are not centred around the true values. Looking again at Table 3.1, we notice that only 10 observations are used in the update of those parameters, as the others are missing and are imputed by the model.

In Figure 3.9, the posterior of the number of clusters is displayed. As one can see, the model provides a good estimate of the number of groups in the trajectories of the patients. The posterior mode, indeed, is located at 2 with a probability greater than 0.8, which can be considered a very precise result.

The posterior distribution of σ, along with the 95% credible intervals, is shown in Figure 3.10. We see that the true $\sigma = 1.0$ is contained in the credible intervals and therefore the result is acceptable.

Let us now focus on the primary parameters of interest when dealing with a regression

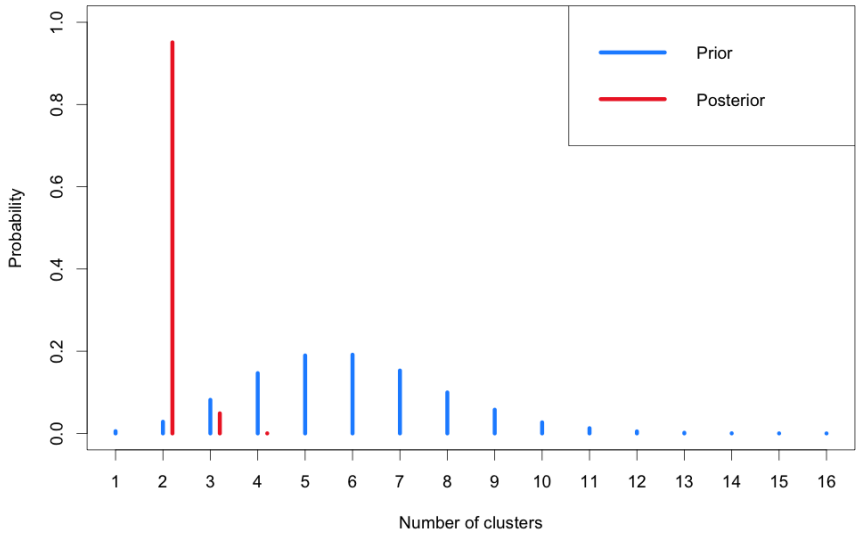


Figure 3.9: Prior and posterior number of clusters, i.e. of the number of unique values in the $\boldsymbol{\alpha}_i$*'s.*

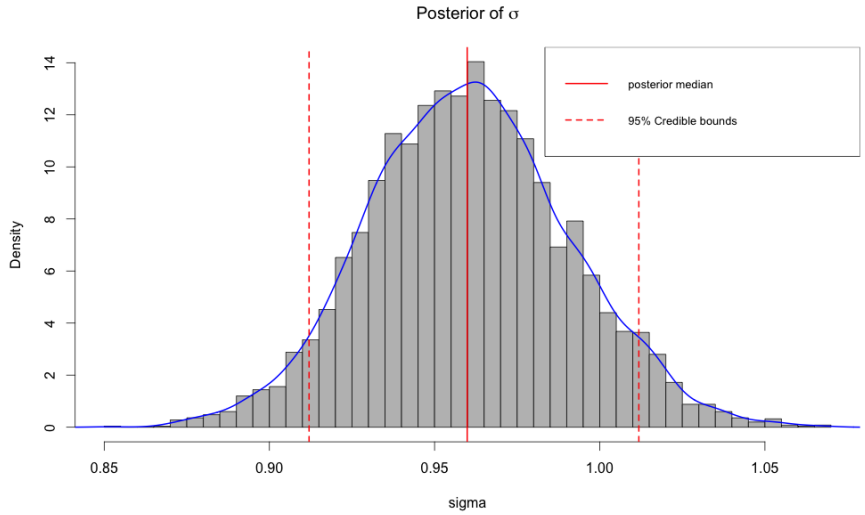
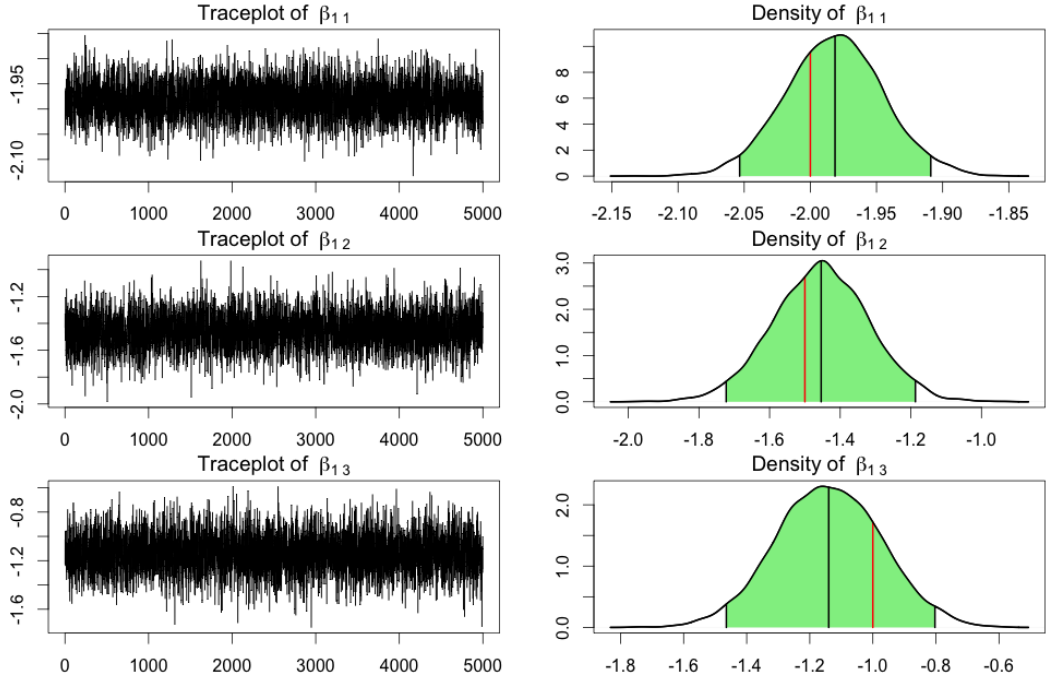


Figure 3.10: In blue solid line, the posterior distribution of the parameter σ*, whose true value is* 1.0*. A point estimate (the posterior median) and the* 95% *credible bounds are overlaid in red.*

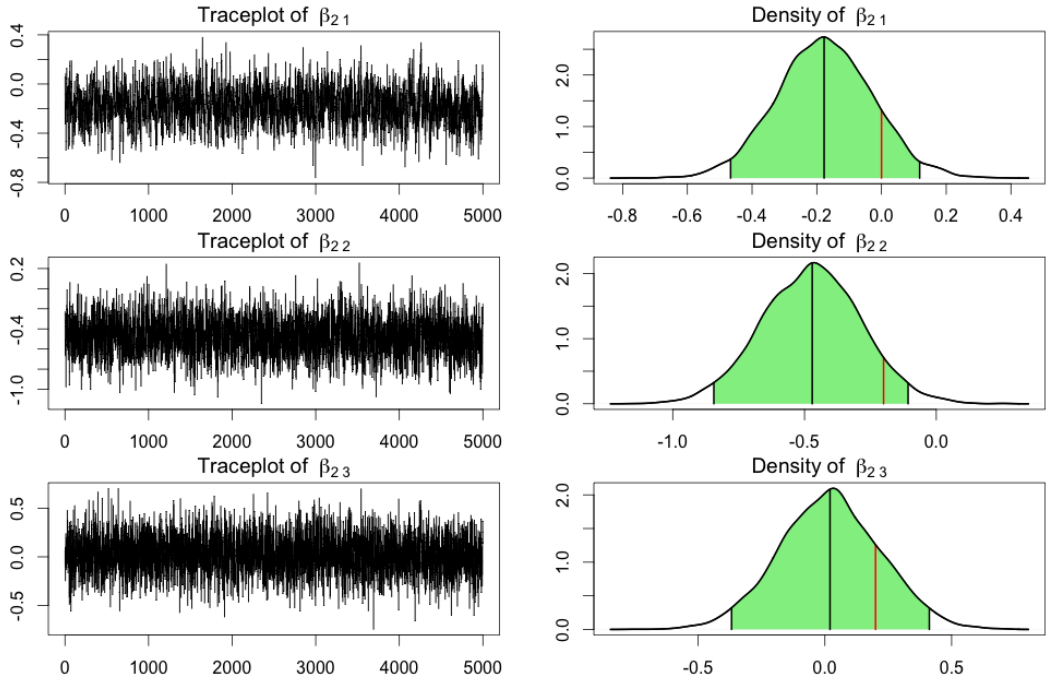
model, i.e. the covariate parameters $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J$. In Figure 3.11 the traceplots and the density estimates of those parameters are reported. First of all, from the traceplots we cannot exclude that the MCMC chain has reached its stationary distribution. Moreover, as one can see, the estimates are correct.

Looking at the posterior distributions, one can remark that the estimates of the covariate parameters concerning the first gap times are more precise than ones related to

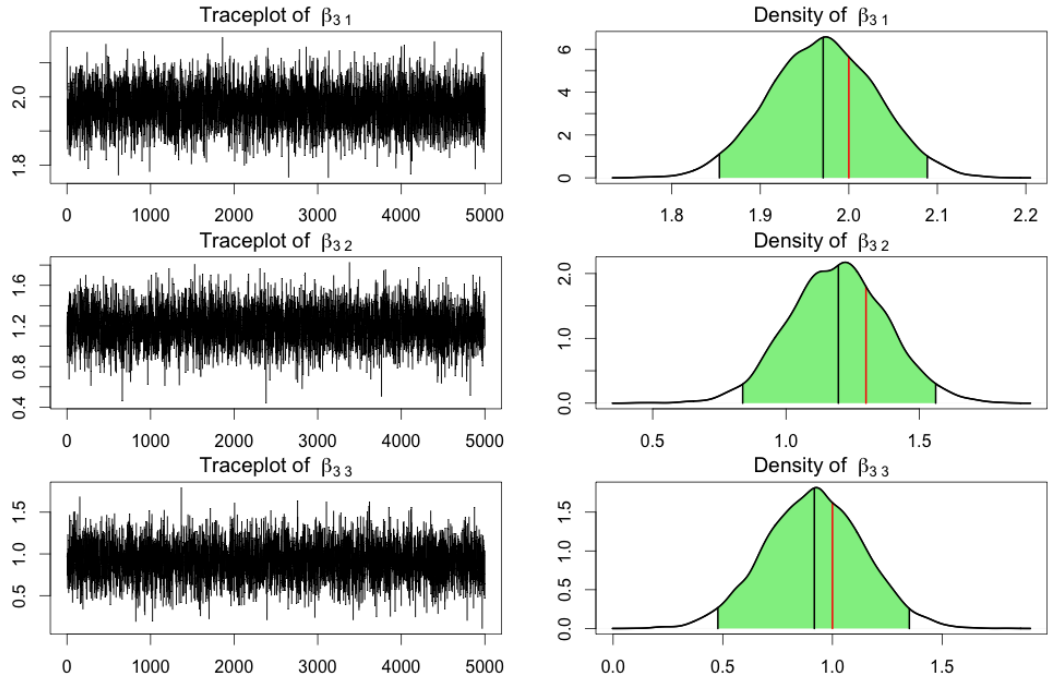
the last ones. The reason is the same as before for the $\boldsymbol{\alpha}$ parameters: fewer observations are involved in the update of the last parameters because we do not observe for all the data J recurrent events.



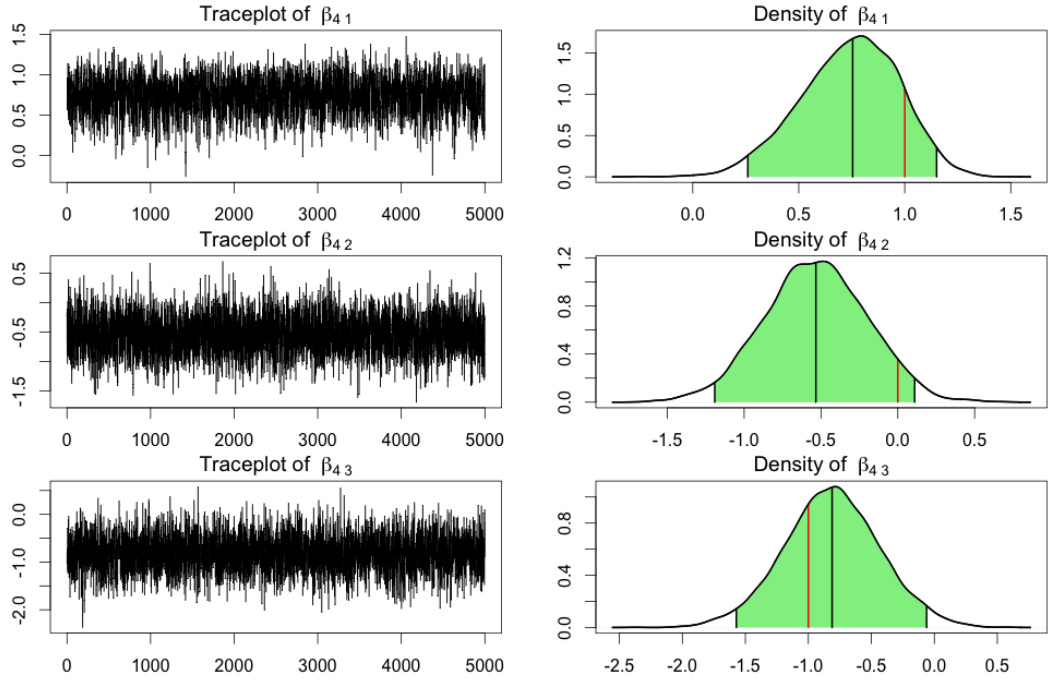
(a) Traceplot and posterior density estimate of $\boldsymbol{\beta}_1$.



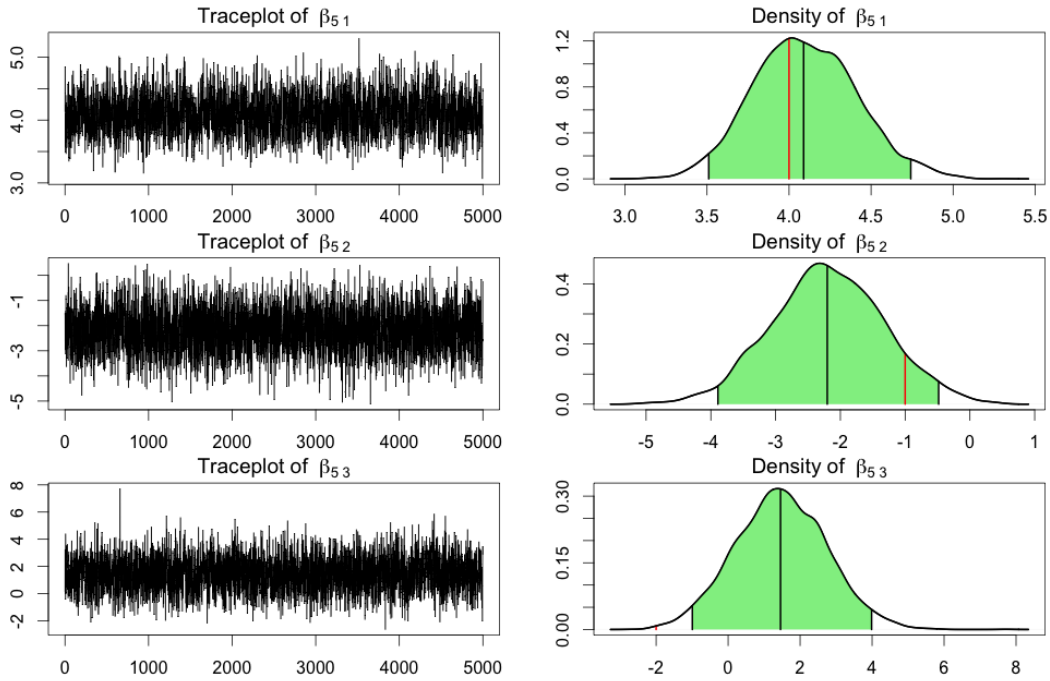
(b) Traceplot and posterior density estimate of $\boldsymbol{\beta}_2$.



(c) Traceplot and posterior density estimate of $\boldsymbol{\beta}_3$.



(d) Traceplot and posterior density estimate of $\boldsymbol{\beta}_4$.



(e) Traceplot and posterior density estimate of $\boldsymbol{\beta}_5$.

Figure 3.11: Traceplot and posterior density estimates of the covariate parameters $\boldsymbol{\beta}_i$. The green shadowed area represents the 95% credible interval, and the vertical black solid line the posterior median. The vertical red solid lines are the true values from which the data have been generated.

3.5 Possible extensions and modifications

The first extension of the model proposed in this chapter is the introduction of censored data. The missing observation of the last gap time has to be imputed, and its contribution to the total likelihood has to be included in the model, as done in Section 4.2.1.

Furthermore, another level of hierarchy can be added in order to infer on the total mass parameter M and to make the model less sensitive with respect to its choice.

A second idea is to provide a time-dependent clustering structure for the observations. In our model, a global clustering is obtained, i.e. we get the same clustering structure and number of clusters at each time interval because the entire trajectories are considered. It is probable that this structure leads to the creation of many clusters, especially when J is large, as the observations are likely to have a lot of different behaviours with respect to their entire event histories.

Therefore, accordingly to the conditional structure of our model (the joint vector of the gap times for each observation is decomposed in the product of the conditional laws), we could link the base measures G_{0j} $\forall j = 1, \ldots, J$ in order to model each gap time separately by keeping the dependence on the previous ones. Alternatively, if the base measures are given independently, the total mass parameters can be linked in order to let the gap times exchange some information.

If such a time-varying clustering structure is obtained, the aim of the analysis would be to inspect the variation of the partitions in the observations. In fact, let us suppose that, in a medical application, a patient changes cluster between the j^{th} and the $(j+1)^{th}$ event. That would mean that there is evidence for an effect of the j^{th} event (e.g. the hospitalisation).

Chapter 4

Application to patients diagnosed with colorectal cancer

In this chapter, we extend the analysis of the model proposed in Chapter 3 to a real dataset. In particular, we use here the *readmission* dataset in the *frailtypack* package of the statistical software R. We also compare the results with the "shared frailty model", which is a semiparametric method used to estimate the hazard function when the observations belong to different clusters. Since a great proportion of the observations is censored, both models take into account the right-censoring of the data.

4.1 Introduction

4.1.1 The dataset

In this chapter we use the *readmission* dataset in the *frailtypack* package of R. This dataset contains rehospitalisation times (in days) after surgery in patients diagnosed with colorectal cancer.

In Figure 4.1 a preview of the data is displayed. In this dataset $n = 403$ patients are available, for a total number of 861 recurrent events. Available data for each patient are:

- *id*: identification code of each subject.
- *time*: gap time since the previous event.
- *event*: rehospitalisation status. This variable takes the value 1 for each subject with the exception of the last event.
- *chemo*: variable indicating if the patient received chemotherapy.
- *sex*: gender of the patients.

- *dukes*: variable indicating the classification of the colorectal cancer. The cancer is more and more severe as this variable augments: the baseline A-B denotes the invasion of the tumour through the bowel wall penetrating the muscle layer but not involving lymph nodes; the value C indicates the involvement of lymph nodes; the value D implies the presence of widespread metastases.

- *charlson*: Charlson comorbidity index. In medicine, comorbidity describes the effect of all other diseases an individual patient might have other than the primary disease of interest. This index measures the ten-year mortality for a patient who may have a range of comorbid conditions: the possible values are 0, 1-2, and 3.

- *death*: binary variable indicating if the patient survived or not.

id	enum	t.start	t.stop	time	event	chemo	sex	dukes	charlson	death
1	1	0	24	24	1	Treated	Female	D	3	0
1	2	24	457	433	1	Treated	Female	D	0	0
1	3	457	1037	580	0	Treated	Female	D	0	0
2	1	0	489	489	1	NonTreated	Male	C	0	0
2	2	489	1182	693	0	NonTreated	Male	C	0	0
3	1	0	15	15	1	NonTreated	Male	C	3	0
3	2	15	783	768	0	NonTreated	Male	C	3	1
4	1	0	163	163	1	Treated	Female	A-B	0	0
4	2	163	288	125	1	Treated	Female	A-B	0	0
4	3	288	638	350	1	Treated	Female	A-B	0	0
4	4	638	686	48	1	Treated	Female	A-B	0	0
4	5	686	2048	1362	0	Treated	Female	A-B	0	0
5	1	0	1134	1134	1	NonTreated	Female	C	0	0
5	2	1134	1144	10	0	NonTreated	Female	C	3	0

Figure 4.1: Preview of the dataset: the first five observations are displayed.

The outcome variables in this study are readmission times, considering them as potential recurrent events (colorectal cancer patients may have several readmissions after first discharge). The first readmission time has been considered as the time between the date of the surgical procedure and the first readmission to hospital related to colorectal cancer.

j	1	2	3	4	5	6	TOT
n_j	30	96	36	18	9	8	197

Table 4.1: Number of observations with exactly j gap times, $j = 1, \ldots, J$.

4.1.2 Descriptive analysis

First of all, patients with more than 6 events were crossed out. Thus, $n = 197$ observations for a total number of 495 recurrent events, were obtained.

In Table 4.1 the number of observations with exactly j gap times are shown $\forall j = 1, \ldots, J$. Let us remark that 119 observations out of 197 are right-censored with respect to their last gap time. Since the proportion of censored data is considerable, we have to take them into account as a special case in the algorithm. We will detail this procedure in Section 4.2.1.

In Table 4.2 a cross-table containing the frequency distribution of the covariates with respect to the sex of the patients can be used to determine whether there is a relation between sex and the other covariates.

	Men n (%)	**Women** n (%)	p-value
Dukes stage			
A-B	49 (0.42%)	27 (0.38%)	
C	44 (0.38%)	35 (0.48%)	
D	29 (0.20%)	13 (0.14%)	0.2992
Chemotherapy			
Yes	62 (0.50%)	44 (0.39%)	
No	60 (0.50%)	31 (0.61%)	0.3547
Charlson Index			
0	221 (0.70%)	127 (0.71%)	
1-2	25 (0.08%)	6 (0.04%)	
3	70 (0.22%)	46 (0.25%)	0.1095

Table 4.2: Contingency table containing the frequency distribution of the covariates with respect to the sex of the patients. In the last column, the p-value of the χ^2-test of independence is calculated.

Pearson's χ^2-test of independence, whose results are displayed in Table 4.2, is a classical test in the framework of count data. The p-values show that there does not exist a statistical dependence between sex and the other covariates. In other words, there is no evidence to suggest that men and women tend to have different values of the other covariates.

4.2 Application of the model

Let us now rewrite the model proposed in Chapter 3 adapting it to the *readmission* dataset. The model is

$$\begin{aligned} Y_{i1} &= \boldsymbol{x}_{i1}^T \boldsymbol{\beta}_1 + \alpha_{i12} + \sigma\epsilon_{i1} \\ Y_{ij} &= \boldsymbol{x}_{ij}^T \boldsymbol{\beta}_j + \alpha_{ij1} Y_{i,j-1} + \alpha_{ij2} + \sigma\epsilon_{ij} \quad \forall j = 2, \dots, n_i \end{aligned}$$

where, in this case

$$\boldsymbol{x}_{ij}^T \boldsymbol{\beta}_j = x_{ij1}\beta_{j1} + x_{ij2}\beta_{j2} + x_{ij3}\beta_{j3} + x_{ij4}\beta_{j4} + x_{ij5}\beta_{j5} + x_{ij6}\beta_{j6}.$$

The components of each vector $\boldsymbol{x}_j$ $\forall j = 1, \dots, J$ are the following:

- x_{j1} is a dummy variable that is equal to 1 if the patient received chemotherapy treatment during gap time j (baseline: not received);
- x_{j2} is a dummy variable denoting the sex of the patient during gap time j. It is equal to 1 if the patient is a woman (baseline: man);
- x_{j3} is a dummy variable indicating the stage of the tumour, measured by Dukes index, during gap time j. It is equal to 1 if the tumour is at stage C (baseline: A-B);
- x_{j4} is a dummy variable indicating the stage of the tumour, measured by Dukes index, during gap time j. It is equal to 1 if the tumour is at stage D (baseline: A-B);
- x_{j5} is a dummy variable indicating Charlson index during gap time j. It is equal to 1 if the index is $1-2$ (baseline: 0);
- x_{j6} is a dummy variable indicating Charlson index during gap time j. It is equal to 1 if the index is 3 (baseline: 0).

Only the last two covariates, i.e. Charlson index, are time-dependent in this framework. However, we allow each covariate to have a different effect according to the current gap time. Therefore, the model is

$$\begin{aligned} &\boldsymbol{Y}_i | \boldsymbol{\alpha}_i, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_J, \boldsymbol{x}_i, \sigma^2 \overset{\text{ind}}{\sim} k(\boldsymbol{y}; \boldsymbol{\alpha}_i, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_J, \boldsymbol{x}_i, \sigma^2) = \mathcal{N}_J(\boldsymbol{\mu}_i, \Sigma_i) \\ &\sigma^2 \sim \text{inv-gamma}\left(\frac{\nu_0}{2}, \frac{\nu_0 \sigma_0^2}{2}\right) \\ &\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_J \overset{\text{iid}}{\sim} \mathcal{N}_p(\mathbf{0}, \beta_0^2 \mathbb{I}_p) \\ &\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_n | G \overset{\text{iid}}{\sim} G \\ &G \sim \text{DP}(MG_0) \end{aligned}$$

along with the specification of the base measure

$$
\begin{aligned}
&G_0 = W_1 \otimes Z_2 \otimes W_2 \otimes \cdots \otimes Z_J \otimes W_J, \\
&W_1, \dots W_J \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_W^2), \\
&Z_2, \dots Z_J \overset{\text{iid}}{\sim} \text{Uniform}(a_Z, b_Z).
\end{aligned}
$$

Let us recall that the quantities $\boldsymbol{\mu}_i, \Sigma_i$ depend on both $\boldsymbol{\alpha}$ and σ^2, and are defined in Section 3.1.

4.2.1 Introducing censored and missing data

In the following, we detail the procedure in order to include censored data and missing data in the model. First, let us remark that right-censoring is present in the dataset only for the last observed gap time. Therefore one knows that the n_i^{th} gap time of patient i is larger than a certain value, i.e. $T_{n_i} \geq \tau_i$ where τ_i is the final observation time, and this information has to be considered in the study.

A simple strategy to deal with this complication is represented by an augmentation of the state space that allows us to include also the "true" (and unknown) gap times. The augmented Gibbs sampling strategy is straightforward.

Let us denote, for the sake of brevity, with $\boldsymbol{\theta}$ the whole parameter vector, and with $\boldsymbol{z}$ the vector of the unknown last gap times that are censored. It is then sufficient to add a full conditional to the model proposed above. In fact, in order to sample from $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{y})$ one can sample recursively from:

- $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{z})$, which is the same set of full conditionals described in Section 3.2.2, where now the censored gap times have been replaced by the simulated "observed" gap times;
- $\mathcal{L}(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{\theta})$, which is a new full conditional.

The latter is easy to determine. In fact, if the censoring information concerning the last gap times is to be included (i.e. if we condition also with respect to Y_{n_i}), then the gap time Z_i can be sampled via

$$
Z_i|Y_{in_i}, Y_{in_i-1}, \boldsymbol{x}_{in_i}, \boldsymbol{\beta}_{n_i}, \boldsymbol{\alpha}_i, \sigma^2 \sim \mathcal{N}(\boldsymbol{x}_{in_i}^T \boldsymbol{\beta}_{n_i} + \alpha_{in_i 1} Y_{in_i-1} + \alpha_{in_i 2}, \sigma^2) \mathcal{I}_{(y_{in_i}, +\infty)}, \quad (4.1)
$$

which is a truncated normal distribution. Let us remark here that one should impute also the covariates at the "future" and unknown time Z_i. For simplicity, we use here the covariates at the time of the censoring, i.e. $\boldsymbol{x}_{in_i}$.

Therefore, at each iteration, we sample the last gap times corresponding to the censored observations using the current values of the MCMC chain. After that, these values are used in order to estimate the parameters for the next iteration.

The handling of missing gap times is equivalent: we sample, at each iteration of the MCMC algorithm, the gap times until we obtain trajectories of length J. This is done equivalently as in (4.1), with the only difference that the normal distribution is not truncated since we do not have the censoring information. In this case, since some of the covariates are time varying, we cannot use the same covariates for all the unknown gap times. Therefore we also sample the future paths of the covariates under the hypothesis of "missing at random". The R package *mi* was used in order to sample the missing covariates and to lead to more realistic estimates.

4.3 Posterior analysis

We now present the inference corresponding to the application of this model to the *readmission* dataset.

In this section, the hyperparameters of the prior distributions are

$$\begin{aligned}
&\beta_0 = 10;\\
&\nu_0 = 4.02; \quad \sigma_0 = 0.7089;\\
&M = 0.1;\\
&\sigma_W^2 = 100; \quad a_Z = -1; \quad b_Z = 1.
\end{aligned}$$

The hyperparameters related to the base measure G_0 of the DP prior are chosen as non-informative as possible (normal distributions with high variance and uniform distributions over the domain interval for intercept and slope terms, respectively).

As far as the choice of the hyperparameters of σ^2 is concerned, the values are chosen in order to obtain an a priori expected value of 1 and an a priori variance equal to 100. In fact, the following equations

$$\begin{aligned}
\mathbb{E}[\sigma^2] &= \frac{\frac{\nu_0\sigma_0^2}{2}}{\frac{\nu_0}{2}-1} = \frac{\nu_0\sigma_0^2}{\nu_0-2} = 1\\
\mathrm{Var}(\sigma^2) &= \frac{\frac{\nu_0^2\sigma_0^4}{4}}{(\frac{\nu_0}{2}-1)^2(\frac{\nu_0}{2}-2)} = \frac{2\nu_0^2\sigma_0^4}{(\nu_0-2)^2(\nu_0-4)} = 100
\end{aligned}$$

lead to the values described above. Moreover, the model is robust with respect to the specification of the hyperparameters (ν_0, σ_0). In another setting the values $\nu_0 = 2$, $\sigma_0^2 = 1$ were tried, yielding a prior distribution with heavy tails for σ^2 (neither the first nor the second moment exist). In this latter case, the posterior inference results were the same.

Analogously, a high value for the standard deviation of the covariate parameters $\boldsymbol{\beta}_j$ is set. Even in this case a robustness analysis was performed, by setting a even higher value $\beta_0 = 50$. No change in the posterior results could be detected.

Posterior estimates are computed via the collapsed Gibbs sampler algorithm presented in Section 3.2.2. We run the algorithm in Julia for $200,000$ iteration, while the first $50,000$ iterations were discarded and we use a thinning of 30 to reduce the autocorrelation of the Markov chain. The final sample size is then 5000.

4.3.1 Posterior inference on the number of clusters and predictive inference for cluster-specific parameters

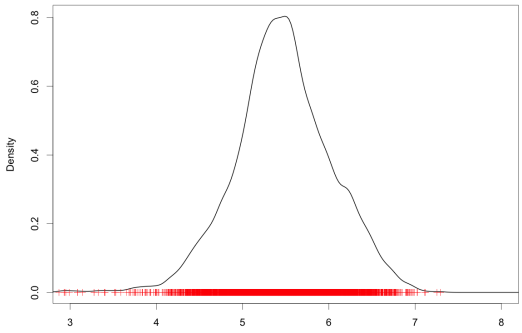
Here the inference on the parameters relative to the different clusters is illustrated. With our model, the posterior mode of the number of clusters is 4 with a probability of around 70%, but also 5 clusters is a plausible value, as shown in Figure 4.3.

In Figure 4.2 the values of the posterior predictive distributions for each component of the cluster specific parameter $\boldsymbol{\alpha}$ are displayed. Let us remark that $J = 6$ corresponds to a 11-dimensional $\boldsymbol{\alpha}$ parameter. However, we here decided to illustrate only the inference on the most significant components, i.e. from the first to the seventh component, as the last ones are updated by a lot of imputed (non-observed) data and therefore they have large variances.

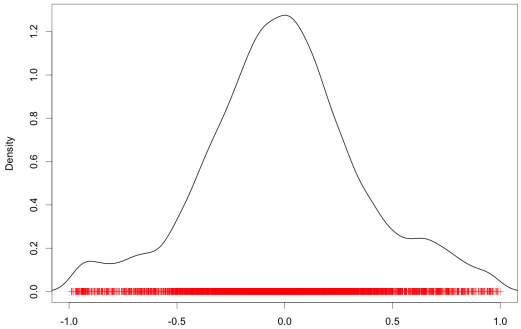
Let us now analyse how the observations are clustered together according to their entire trajectories. Let us recall that the optimal partition was calculated via the loss function method presented in 3.2.3. In Figure 4.4 the trajectories and the recurrent events of the patients are displayed, coloured according to their cluster labels.

Let us try to give a qualitative description of the clustering structures. As one can see, the first group of patients is the largest, and it is characterised by a few recurrent events (2 or 3). The second group mainly consists of patients having a high number of recurrent events. The third group, instead, puts together patients with a lot of events occurring rapidly. The last two groups exhibit different behaviours.

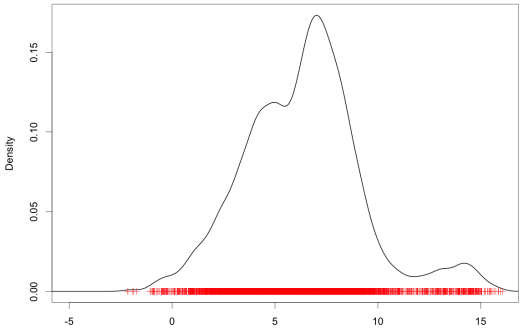
In Section 4.5 we provide a robustness analysis for the clustering structure with respect to the choice of the DP prior, and we see how the optimal partition changes accordingly.



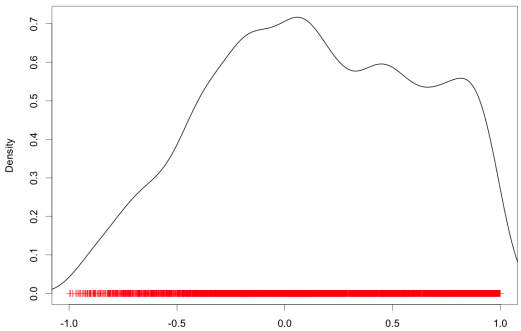
(a) Predictive density of α_{12}.



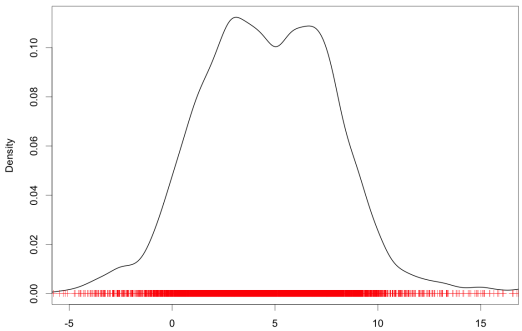
(b) Predictive density of α_{21}.



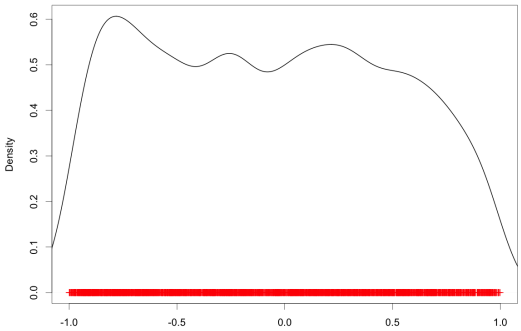
(c) Predictive density of α_{22}.



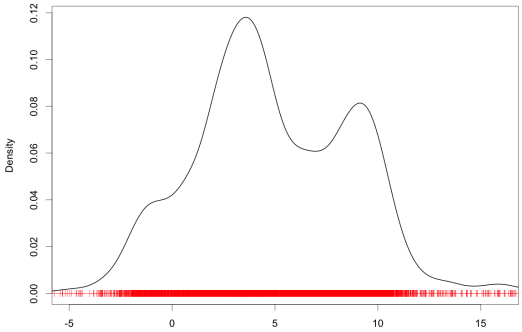
(d) Predictive density of α_{31}.



(e) Predictive density of α_{32}.



(f) Predictive density of α_{41}.



(g) Predictive density of α_{42}.

Figure 4.2: In black solid line, kernel density estimates of the predictive distributions of the $\boldsymbol{\alpha}$ parameters. The red ticks on the x-axis represent the sampled values.

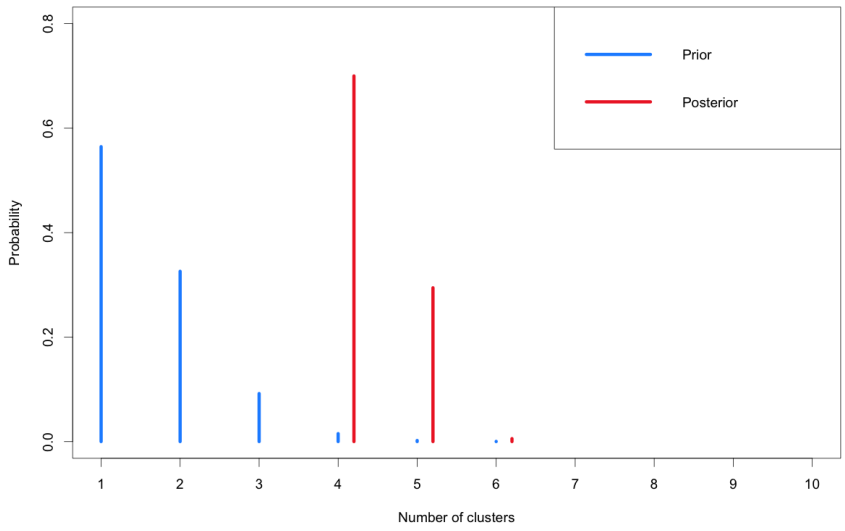


Figure 4.3: Prior and posterior number of clusters, i.e. of the number of unique values in the $\boldsymbol{\alpha}_i$'s.

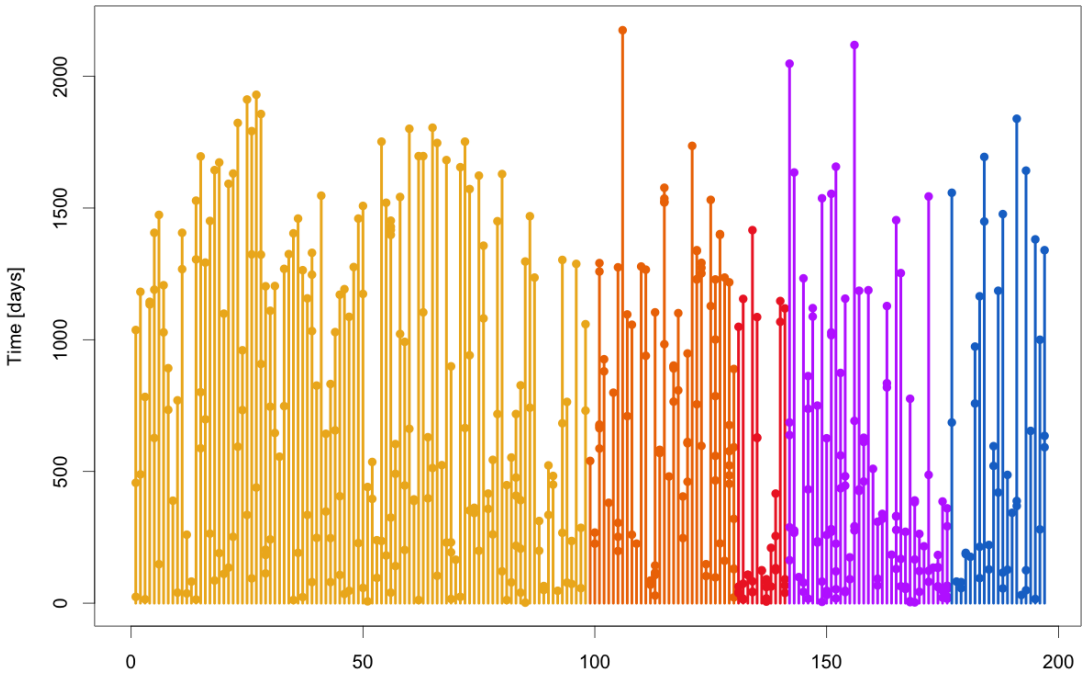


Figure 4.4: In vertical lines, the total length of the paths for each patient. The points represent the recurrent events, and the colours are defined by the cluster labels.

4.3.2 Posterior inference on the regression parameters

We now discuss the inference on the regression parameters in order to understand how covariates influence the recurrent events, regardless of the underlying structure of the trajectories (which is captured by the cluster-specific parameters). In this section, we denote with $\widetilde{\boldsymbol{\beta}}_i$ the vector of the parameter relative to the i^{th} covariate for each gap time. Remark that, on the other hand, we denoted with $\boldsymbol{\beta}_i$ the vector of all the covariate parameters for the i^{th} gap time.

First of all, the convergence of the chain is checked via Geweke's statistics. The idea behind this test is simple: it is analogous to a test for the equality of the means of the first and last part of a Markov chain (by default the first 10% and the last 50%). If the samples are drawn from the stationary distribution of the chain, the two means are equal and Geweke's statistic has an asymptotically standard normal distribution. In Figure 4.5 the test statistics are displayed. Since the values in the interval $[-2, 2]$ in the majority of cases, we can conclude that the MCMC chain is stationary.

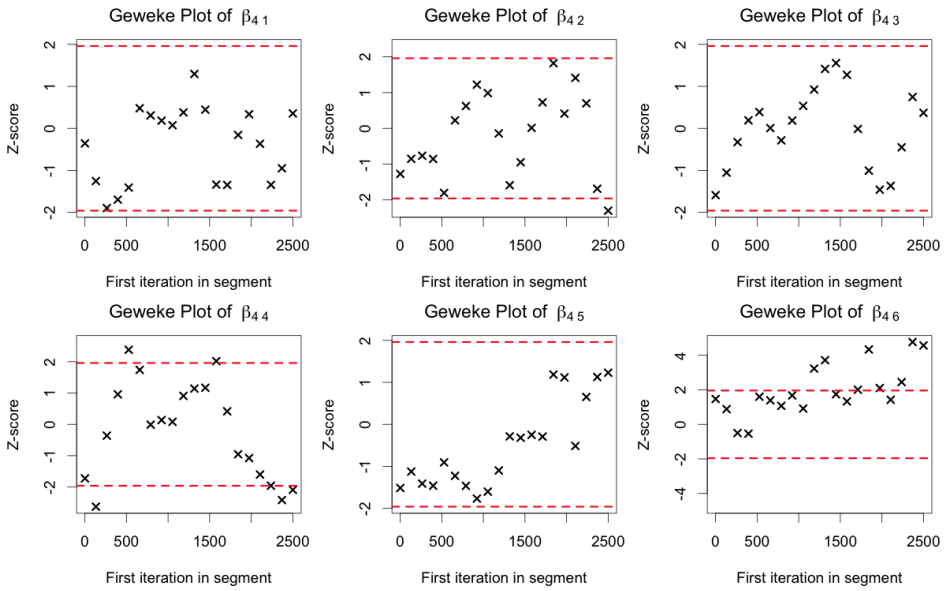
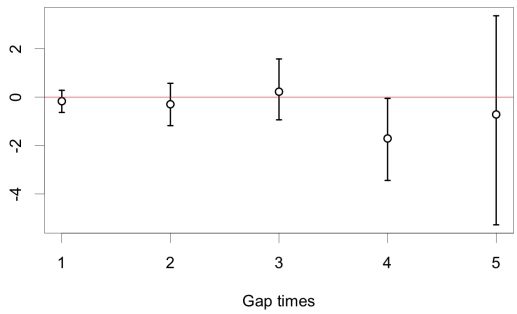


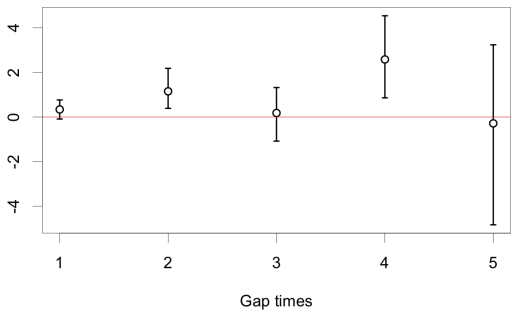
Figure 4.5: Geweke's diagnostic of convergence for $\widetilde{\boldsymbol{\beta}}_4$.

Let us focus on the influence that each covariate has on the outcome variable. By analysing Figure 4.6, which shows the 95% credible intervals for the posterior marginals of the regression parameters, one can deduce the following considerations.

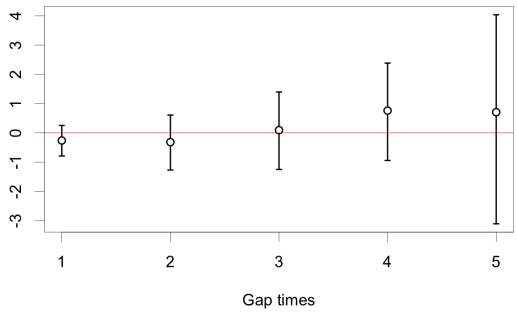
- $\widetilde{\boldsymbol{\beta}}_1$, which captures the effect of the chemotherapy on the gap times, does not seem to be significant for the first gap times. However, at the fourth gap time the CI for β_{14} is concentrated on negative values, which means that chemotherapy reduces



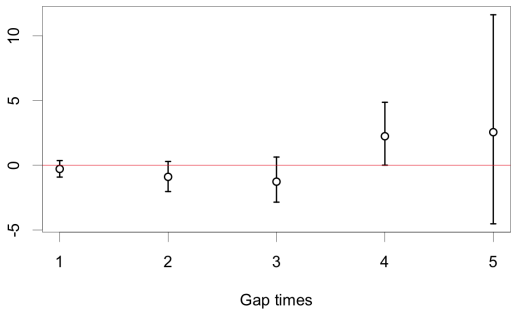
(a) Posterior 95% CI for $\widetilde{\boldsymbol{\beta}}_1$.



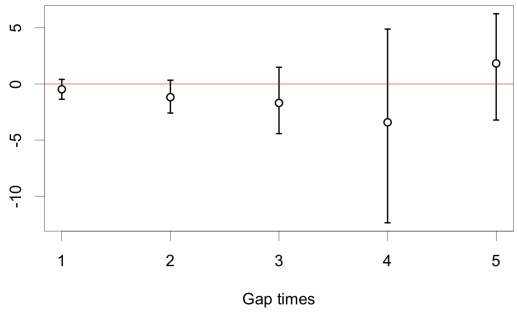
(b) Posterior 95% CI for $\widetilde{\boldsymbol{\beta}}_2$.



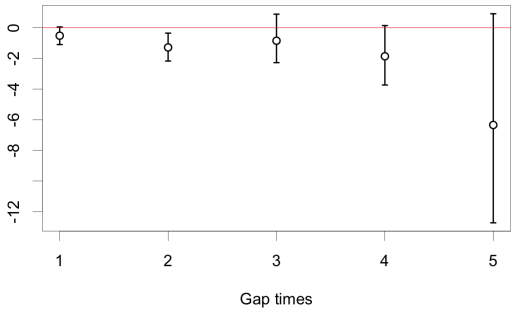
(c) Posterior 95% CI for $\widetilde{\boldsymbol{\beta}}_3$.



(d) Posterior 95% CI for $\widetilde{\boldsymbol{\beta}}_4$.



(e) Posterior 95% CI for $\widetilde{\boldsymbol{\beta}}_5$.



(f) Posterior 95% CI for $\widetilde{\boldsymbol{\beta}}_6$.

Figure 4.6: Posterior 95% credible bounds for each covariate $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_p$ as a function of the gap times.

the fourth waiting time between hospitalisations, i.e. the time elapsed between the fourth and the fifth recurrent events. In general, however, there does not seem to be an effect of chemotherapy on the outcome.

- $\widetilde{\boldsymbol{\beta}}_2$, which measures the effect of sex on the gap times, indicates that women have mainly larger waiting times. This is more evident for the second and the fourth component, but a trend is visible at each recurrent event. This result is not surprising: in fact, this dataset has been originally used in order to find evidence of a

disparity of treatment between women and men.

- $\widetilde{\boldsymbol{\beta}}_3$, the first dummy variable relative to the Dukes stage of the tumour (stage C versus the baseline stage A-B), is never significantly different from 0.

- $\widetilde{\boldsymbol{\beta}}_4$, the second dummy variable relative to the Dukes stage of the tumour (stage D versus the baseline stage A-B), is negative for the gap times from 1 to 3 and positive for the two last gap times. Therefore, seriously ill patients (stage D represents the most advanced stage of the tumour) present early and frequent hospitalisations at the beginning of the study, followed by an opposite effect (delayed hospitalisations).

- $\widetilde{\boldsymbol{\beta}}_5$, the first dummy variable relative to the Charlson Index of the patient (index 1-2 versus the baseline index 0), has negative medians, apart from the last gap time, which is updated by a few data as one can see in Table 4.3. Therefore patients with index 1-2 will experience more frequent recurrent events with respect to the ones with index 0.

- $\widetilde{\boldsymbol{\beta}}_6$, the second dummy variable relative to the Charlson Index of the patient (index 3 versus the baseline index 0), is mostly negative (significantly at gap times 2 and 4). Therefore patients with index 3 will have shorter gap times with respect to the ones with index 0.

Let us remark that, in general, credibility intervals are larger for the last gap times. In fact, the variance of the Gibbs step is increased by the presence of more and more missing gap times.

4.4 Comparison with existing models

Our model is now compared to one of the most popular models in literature: the shared frailty model. This model is usually used as a tool for handling multivariate data in the presence of censoring. Basically, frailty models in this context are random effects models, analogous to those well known from linear normal model theory. However, the frailty models are better adapted to handle censored data than the normal models. The dependence is modelled through a frailty variable, such that all gap times that are related to each other in the same observation have the same level of frailty attached to them.

Let us denote with W_{ij} the recurrent events, with C_{ij} the right-censored times and with L_{ij} the left-censoring times. Let us define the observations $Y_{ij} = \min\{W_{ij}, C_{ij}\}$ and the censoring indicators $\delta_{ij} = \mathcal{I}_{\{Y_{ij}=C_{ij}\}}$.

	Number of readmissions					
	1	2	3	4	5	6
Sex						
Male	122	103	49	24	12	6
Female	75	64	22	11	5	2
Chemotherapy						
Yes	91	77	29	13	10	5
No	106	90	42	22	7	3
Dukes Stage						
A-B	76	67	28	12	5	1
C	79	70	30	15	8	6
D	42	30	13	8	4	1
Charlson Index						
0	136	117	54	27	11	3
1-2	14	10	3	1	2	1
3	47	40	14	7	4	4

Table 4.3: Contingency table containing the frequency distribution and of the covariates with respect to the number of hospital readmissions.

The hazard function, conditional on the frailty term ω_i, of a shared gamma frailty model for the j^{th} gap time ($j = 1, \ldots, n_i$) in the i^{th} observation ($i = 1, \ldots, n$) is

$$\begin{aligned}
&\lambda_{ij}(t|\omega_i) = \lambda_0(t)\omega_i e^{\boldsymbol{\beta}^T \boldsymbol{x}_{ij}} \\
&\omega_i \overset{\text{iid}}{\sim} \Gamma\left(\frac{1}{\theta}, \frac{1}{\theta}\right) \\
&\mathbb{E}(\omega_i) = 1; \quad \text{Var}(\omega_i) = \theta
\end{aligned}$$

where $\lambda_0(t)$ is the baseline hazard function, $\boldsymbol{\beta}$ the vector of the regression coefficient associated to the covariate vector $\boldsymbol{x}_{ij}$ for the j^{th} gap time in the i^{th} observation.

From a frequentist perspective, this model usually uses a semi-parametric penalized likelihood estimation of the hazard function. The analytical expression of the log-likelihood is

$$\begin{aligned}
l(h_0, \boldsymbol{\beta}, \theta) = \sum_{i=1}^{n} \left[\sum_{j=1}^{n_i} \delta_{ij} \ln(h_{ij}(Y_{ij})) \right] - \left(\frac{1}{\theta} + n_i \right) \ln \left[1 + \theta \sum_{j=1}^{n_i} H_{ij}(Y_{ij}) \right] \\
+ \frac{1}{\theta} \ln \left[1 + \theta \sum_{j=1}^{n_i} H_{ij}(L_{ij}) \right] + \mathcal{I}_{\{n_i \neq 0\}} \sum_{j=1}^{n_i} \ln \left[1 + \theta(n_i - k) \right],
\end{aligned}$$

where $H_0(t)$ is the cumulative baseline hazard function and n_i is the number of recurrent events.

The package *readmission* provides an estimate of the hazard function of such a model. In particular, we obtain the results displayed in Listing 4.1.

```
frailtyPenal(formula = Surv(time, event) ~ cluster(id) + dukes +
    charlson + sex + chemo, data = readmission, cross.validation = TRUE,
    n.knots = 10, kappa = 1, Frailty = TRUE)

  Shared Gamma Frailty model parameter estimates
  using a Penalized Likelihood on the hazard function

                 coef exp(coef) SE coef (H) SE coef (HIH)        z          p
chemoTreated  0.189774  1.208976    0.109905      0.109905  1.72671 8.4220e-02 *
sexFemale    -0.306604  0.735942    0.109955      0.109955 -2.78845 5.2961e-03 ***
dukesC        0.147713  1.159180    0.124164      0.124164  1.18966 2.3418e-01
dukesD        0.436850  1.547824    0.154300      0.154300  2.83117 4.6378e-03 ***
charlson1-2   0.536444  1.709916    0.203579      0.203579  2.63507 8.4119e-03 ***
charlson3     0.581047  1.787910    0.129561      0.129561  4.48473 7.3005e-06 ***

           chisq df global p
dukes    30.0903  2 2.92e-07 ***
charlson 10.9222  2 4.25e-03 ***

    Frailty parameter, Theta: 0.660083 (SE (H): 0.141426 ) p = 1.5256e-06

      penalized marginal log-likelihood = -3243.13
```

Listing 4.1: R output of the frailty model

First of all, let us remark that θ, i.e. the variance of the frailty parameters, is significantly different from 0, which can be shown with a Wald test whose p-value is $1.52 \cdot 10^{-6}$. Thus, the observations are heterogeneous and our clustering analysis is meaningful.

Moreover, the significant covariates are, according to this model, $(\beta_2, \beta_4, \beta_5, \beta_6)$. Remark that a negative coefficient in this hazard model corresponds to a positive coefficient in our model. When $\boldsymbol{\beta} \leq 0$ the hazard function is smaller, which means that the next recurrent event does not happen soon (i.e. the gap time is greater). Therefore the results of the shared frailty model are in agreement with the ones provided by our model. First of all, female patients experience greater gap times. Furthermore, patients more seriously ill (as indicated by high values of Dukes Stage or Charlson Index) have smaller gap times and therefore more recurrent events.

In conclusion, the results of the model proposed in this work agree with existing models in survival analysis. However, specifying a Bayesian non-parametric model provides with a more flexible estimation of the dependence that each hospitalisation has on the following ones. In particular, our model allows us to specify time-specific regression parameters and to inspect the influence of the covariates at each gap time. Moreover, our model provides with a clustering structure by grouping patients that share similar patterns in their trajectories. Furthermore, by modelling the gap times directly (and not by means of the hazard function), the prediction is straightforward and more precise.

4.5 Robustness analysis for the DP prior

In this section a robustness analysis with respect to the choice of the DP prior is carried out. In fact, it is well-known that non-parametric models suffer from the high sensitivity with respect, in particular, to the choice of the base measure G_0.

Test Case	M	Distribution of Z_i's	Distribution of W_i's
A	0.1	Uniform$(-1, 1)$	$\mathcal{N}(0, 10)$
B	1	Uniform$(-1, 1)$	$\mathcal{N}(0, 10)$
C	3	Uniform$(-1, 1)$	$\mathcal{N}(0, 10)$
D	0.1	Uniform$(-1, 1)$	$\mathcal{N}(0, 4)$
E	1	Uniform$(-1, 1)$	$\mathcal{N}(0, 4)$
F	3	Uniform$(-1, 1)$	$\mathcal{N}(0, 4)$
G	0.1	$2 \cdot \text{Beta}(0.5, 0.5) - 1$	$\mathcal{N}(0, 10)$
H	1	$2 \cdot \text{Beta}(0.5, 0.5) - 1$	$\mathcal{N}(0, 10)$
I	3	$2 \cdot \text{Beta}(0.5, 0.5) - 1$	$\mathcal{N}(0, 10)$

Table 4.4: Different settings of hyperparameters of the prior tested for the robustness analysis.

For this reason, we test the model in 9 different configurations of the prior hyperparameters. Both the total mass parameter M and the components of the base measure, i.e. the random variables Z_i's and W_i's, vary in the test cases from **A** to **I**. In particular, the total mass parameter M varies from 0.1 to 3, causing a shift in the prior distribution for the number of clusters. The components of G_0, instead, are chosen to be as non-informative as possible in cases **A** - **C**. They are successively localised in cases **D** - **F** (lower variance for the components W_i's) and in cases **G** - **I** (scaled Beta distribution giving high mass to the extremes of the domain for the components Z_i's).

Note that in experiments **A**, **D** and **G** we fixed $M = 0.1$ so that $\mathbb{E}[K_n] = 1.568$; in cases **B**, **E** and **H** we set $M = 1$ so that $\mathbb{E}[K_n] = 5.841$; in experiments **C**, **F** and **I** we

chose $M = 3$ so that $\mathbb{E}[K_n] = 13.149$.

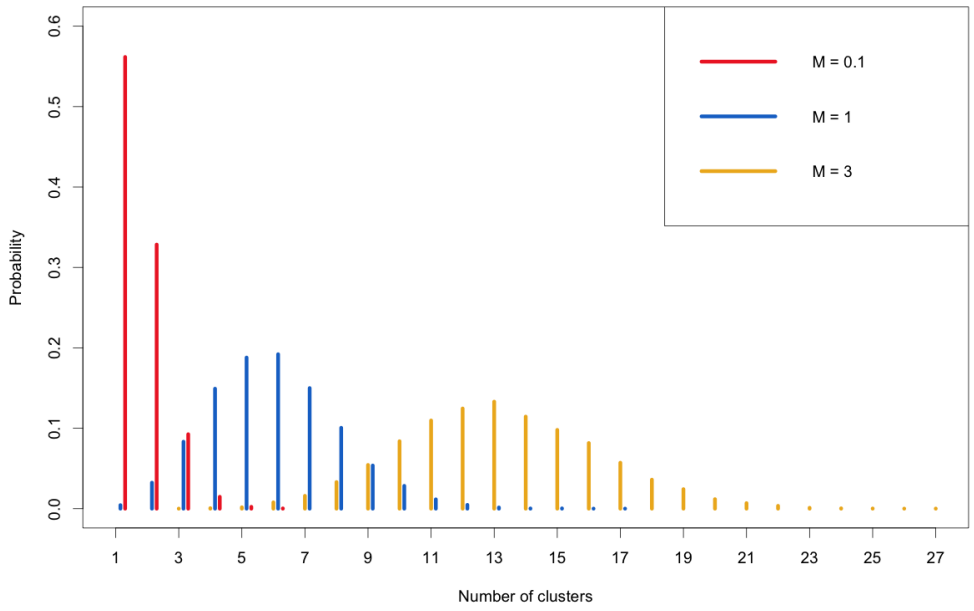


Figure 4.7: Prior distributions of the variable K_n denoting the number of clusters, in cases $M = 0.1$, $M = 1$, and $M = 3$.

Let us also remark that the Dirichlet process induces a prior on the number of clusters which has only one degree of freedom, i.e. the parameter M. Therefore, when increasing the value of M, a double effect is obtained. The prior mode shifts to the right, and the variance increases as well, yielding a flatter distribution, as one can see in Figure 4.7.

Before presenting the results of the robustness analysis, an index for the goodness-of-fit of the model has to be introduced. In this framework, we choose the Log Pseudo Marginal Likelihood (LPML) in order to evaluate the performances of the model and to compare the results obtained with different sets of parameters. The LPML is defined as the sum of the logarithms of the Conditional Predictive Ordinates (CPO) for each observation, i.e.

$$LPML = \sum_{i=1}^{n} \log(CPO_i).$$

CPO_i is the value of the predictive distribution evaluated at y_i, conditioning on the training sample not containing the i^{th} observation, denoted with $\boldsymbol{y}^{(-i)}$. This approach is very common in cross validation techniques, when the data matrix is partitioned in two parts: one is used to estimate the parameters, and the other to measure the goodness of fit. Obviously, the larger the values of the CPO (and, subsequently, of the LPML) the better the model fits the data.

The calculation of LPML consists in the evaluation of n predictive distributions, which can be computationally intense. However, an alternative formula can be proved for $CPO_i = f_i(y_i|\boldsymbol{y}^{(-i)})$. In fact,

$$\begin{aligned} CPO_i = f_i(y_i|\boldsymbol{y}^{(-i)}) &= \int_\Theta f_i(y_i|\boldsymbol{\theta})\mathcal{L}(d\boldsymbol{\theta}|\boldsymbol{y}^{(-i)}) \\ &= \int_\Theta f_i(y_i|\boldsymbol{\theta}) \frac{\prod_{j\neq i} f_j(y_j|\boldsymbol{\theta})\mathcal{L}(d\boldsymbol{\theta})}{\int_\Theta \prod_{j\neq i} f_j(y_j|\boldsymbol{\theta})\mathcal{L}i(d\boldsymbol{\theta})}, \end{aligned}$$

where we used Bayes' theorem. Therefore

$$\begin{aligned} CPO_i^{-1} = \frac{\int_\Theta \prod_{j\neq i} f_j(y_j|\boldsymbol{\theta})\mathcal{L}(d\boldsymbol{\theta})}{\int_\Theta \prod_{i=1}^n f_j(y_j|\boldsymbol{\theta})\mathcal{L}(d\boldsymbol{\theta})} &= \int_\Theta \frac{1}{f_i(y_i|\boldsymbol{\theta})} \frac{\prod_{i=1}^n f_i(y_i|\boldsymbol{\theta})\mathcal{L}(\boldsymbol{\theta})d\boldsymbol{\theta}}{\int_\Theta \prod_{i=1}^n f_j(y_j|\boldsymbol{\theta})\mathcal{L}(d\boldsymbol{\theta})} \\ = \int_\Theta \frac{1}{f_i(y_i|\boldsymbol{\theta})}\mathcal{L}(d\boldsymbol{\theta}|\boldsymbol{y}) \simeq \frac{1}{G}\sum_{g=1}^G \frac{1}{f_i(y_i|\boldsymbol{\theta}^{(g)})}, \end{aligned}$$

where G is the number of iterations and $\boldsymbol{\theta}^{(g)}$ is the value of the chain at iteration g.

In Table 4.5, values of the LPML index for every test are listed.

Test Case	**LPML**
A	−775.34
B	−762.03
C	−748.30
D	−764.18
E	−747.32
F	−745.64
G	−754.30
H	−752.02
I	−745.92

Table 4.5: LPML values in each test case.

Observe that a more complex model will usually be able to better explain the data, and subsequently it will produce a higher value of LPML. In fact, it is clear from Table 4.5 that this index depends on the choice of M: in the test cases when more clusters are provided, the values of LPML are higher.

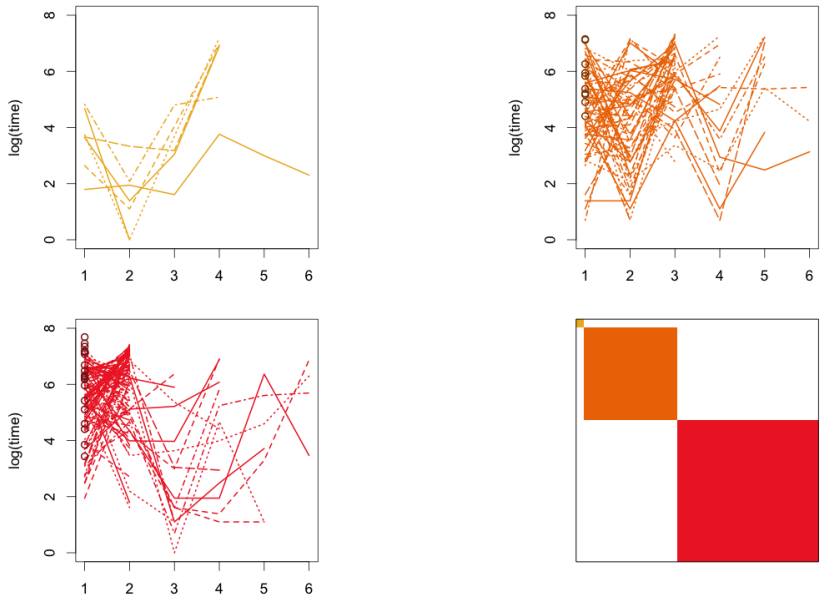


Figure 4.8: Trajectories of the clustered data for test case ***A****. Lines represent the observations with more than one gap time. Points represent the observations with only one event. The incidence matrix is represented in the bottom-right box.*

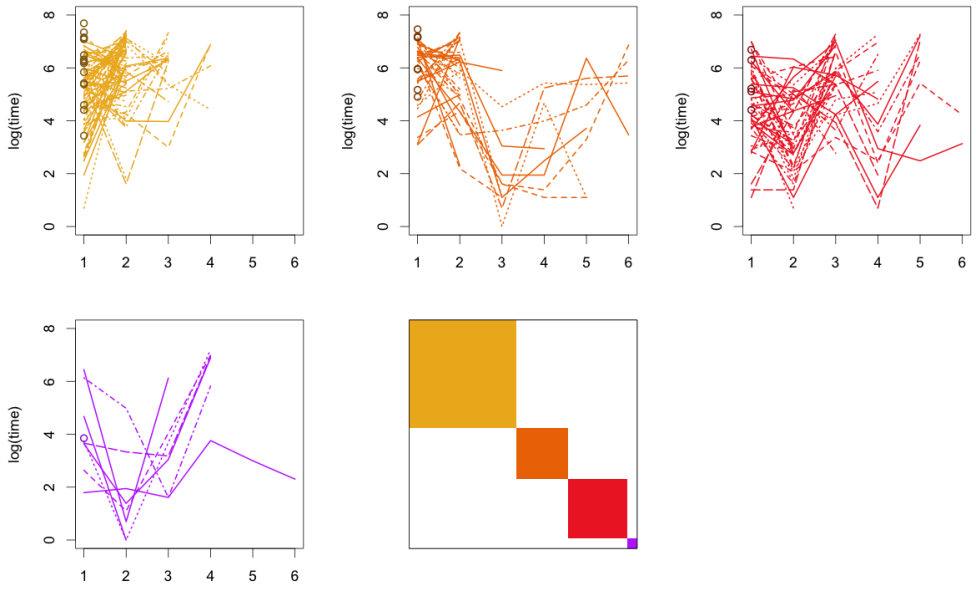


Figure 4.9: Trajectories of the clustered data for test case ***B****. Lines represent the observations with more than one gap time. Points represent the observations with only one event. The incidence matrix is represented in the bottom-right box.*

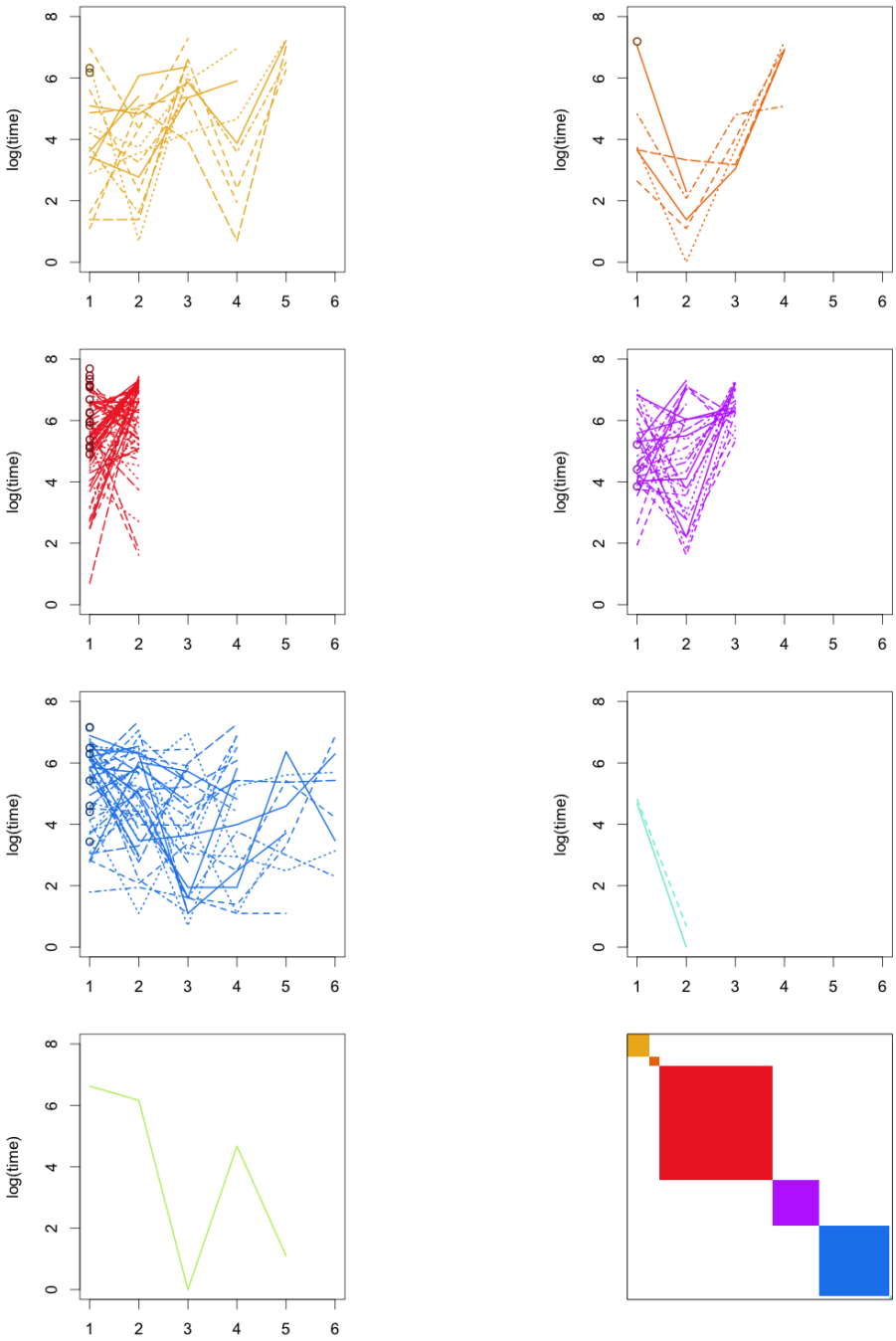


Figure 4.10: Trajectories of the clustered data for test case ***C****. Lines represent the observations with more than one gap time. Points represent the observations with only one event. The incidence matrix is represented in the bottom-right box.*

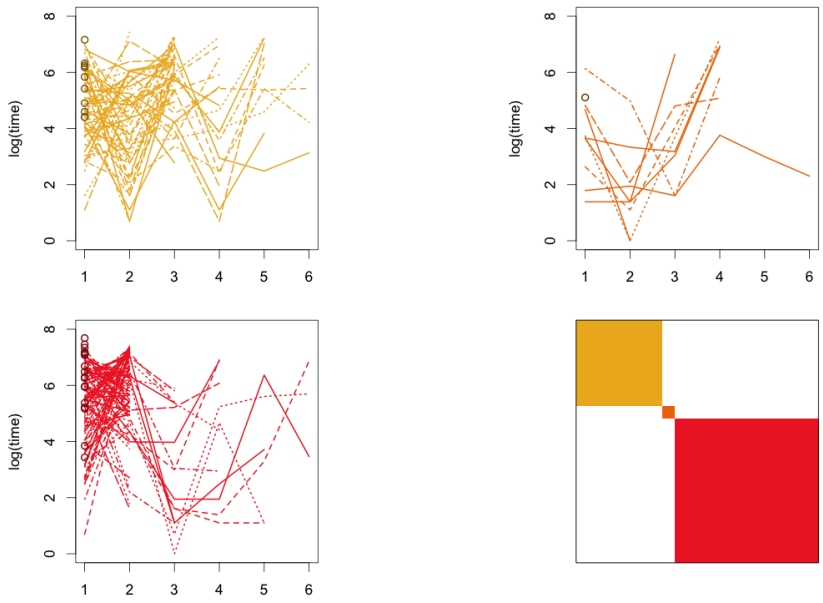


Figure 4.11: Trajectories of the clustered data for test case ***D****. Lines represent the observations with more than one gap time. Points represent the observations with only one event. The incidence matrix is represented in the bottom-right box.*

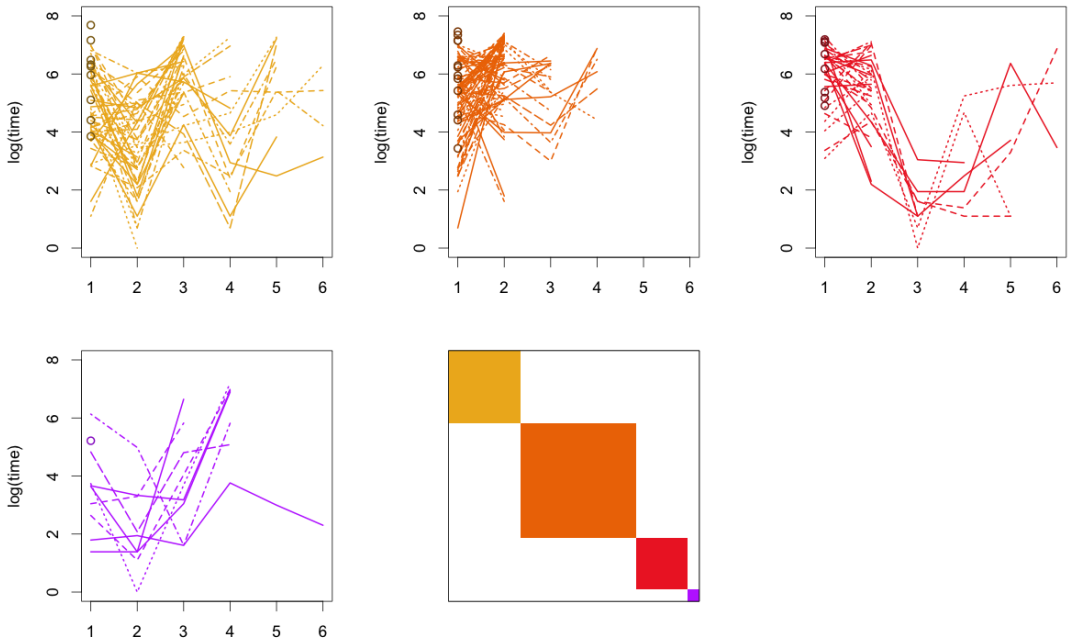


Figure 4.12: Trajectories of the clustered data for test case ***E****. Lines represent the observations with more than one gap time. Points represent the observations with only one event. The incidence matrix is represented in the bottom-right box.*

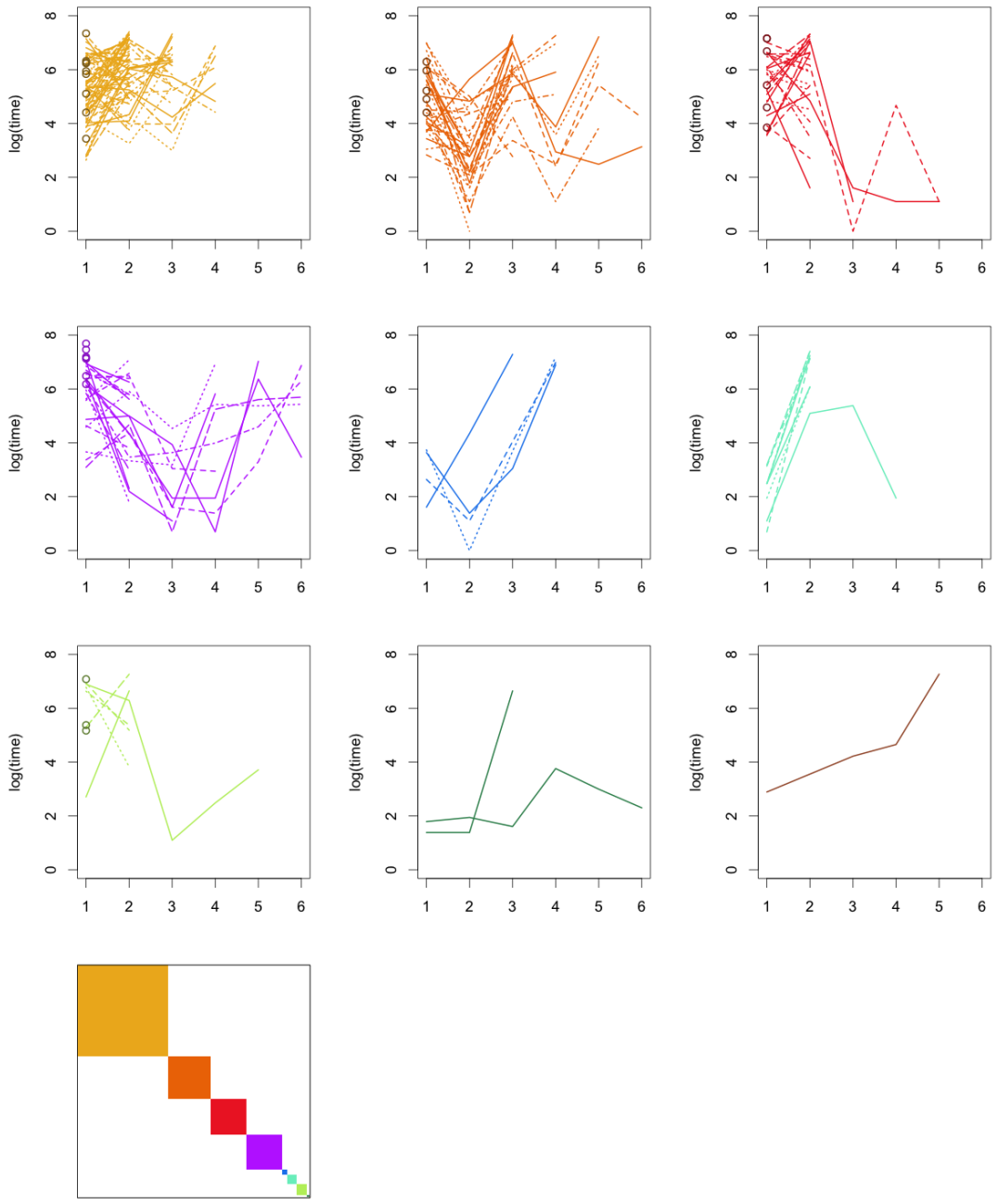


Figure 4.13: Trajectories of the clustered data for test case ***F****. Lines represent the observations with more than one gap time. Points represent the observations with only one event. The incidence matrix is represented in the bottom-right box.*

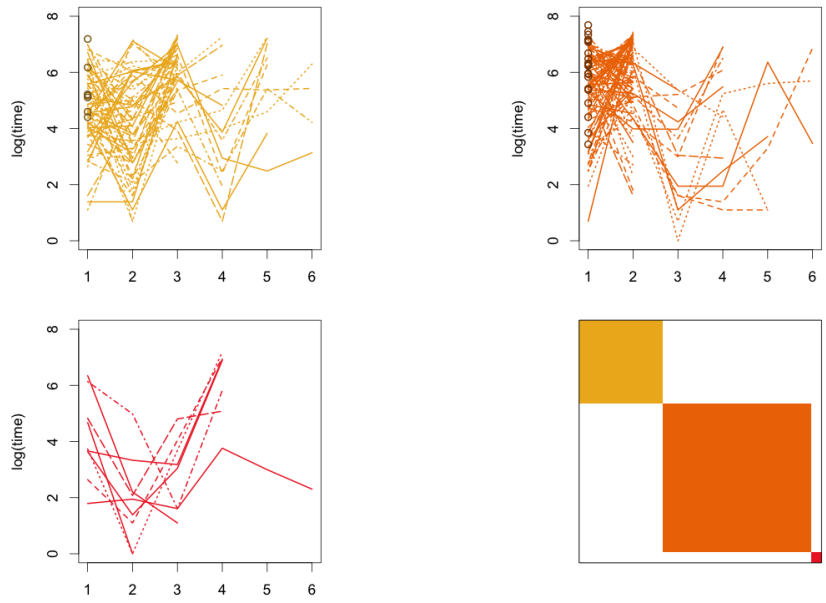


*Figure 4.14: Trajectories of the clustered data for test case **G**. Lines represent the observations with more than one gap time. Points represent the observations with only one event. The incidence matrix is represented in the bottom-right box.*

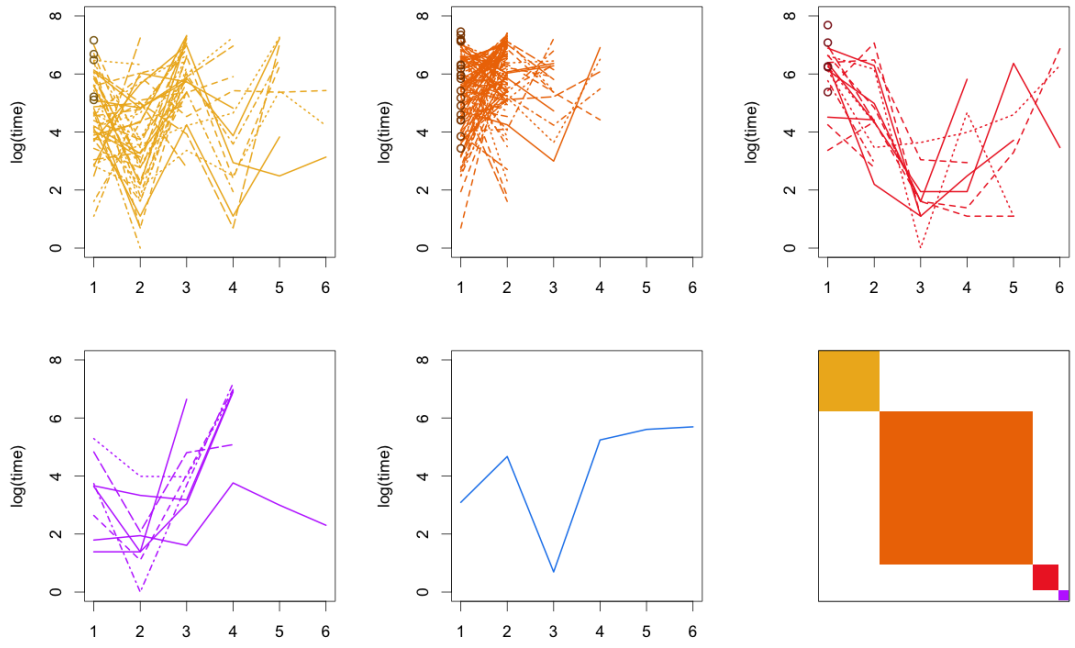


*Figure 4.15: Trajectories of the clustered data for test case **H**. Lines represent the observations with more than one gap time. Points represent the observations with only one event. The incidence matrix is represented in the bottom-right box.*

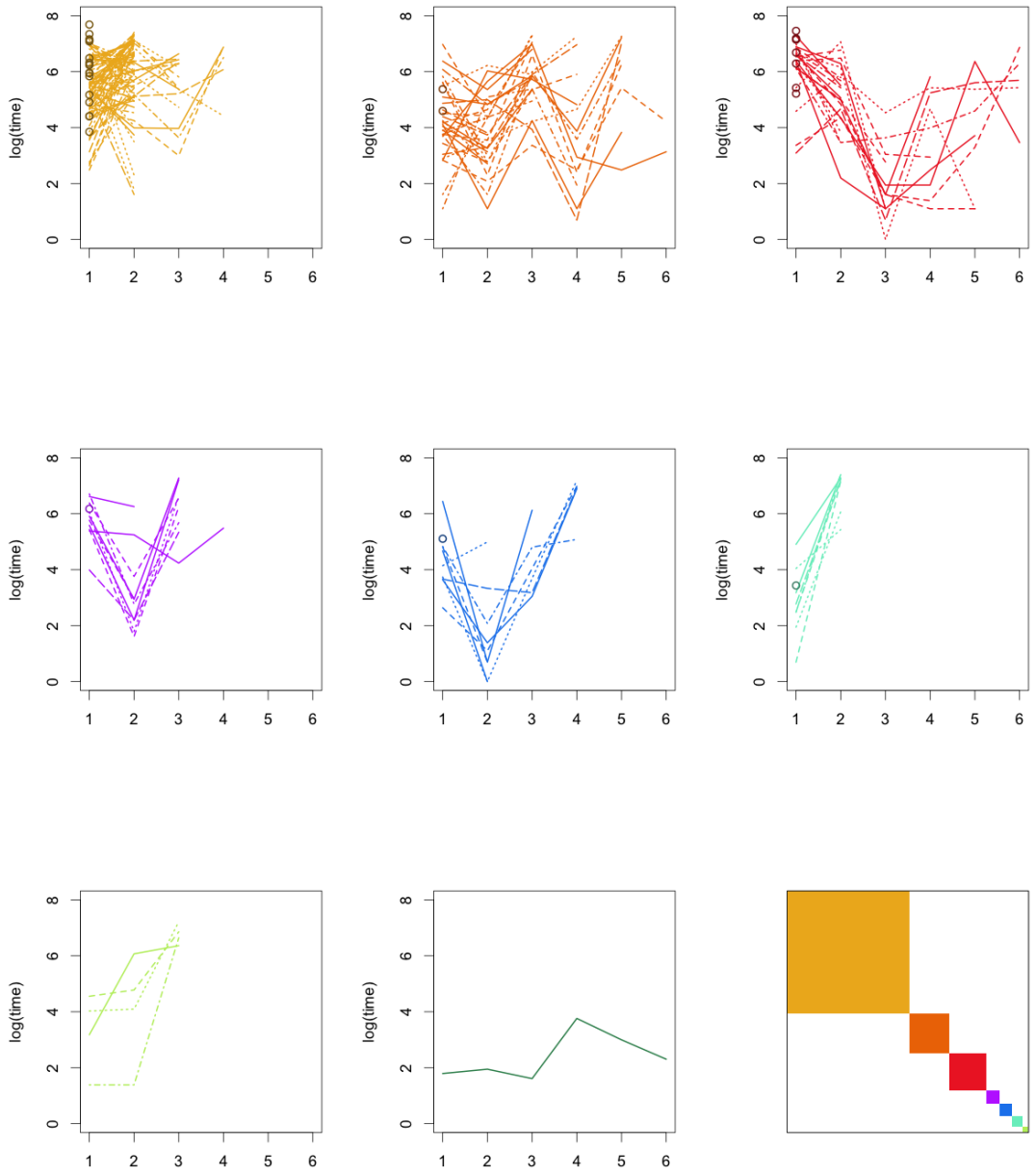


*Figure 4.16: Trajectories of the clustered data for test case **I**. Lines represent the observations with more than one gap time. Points represent the observations with only one event. The incidence matrix is represented in the bottom-right box.*

From Figure 4.8 to Figure 4.16 the trajectories of the data for each cluster are displayed, according to the prior settings specified in Table 4.4. Again, it is evident that the choice of M strongly influences the posterior number of clusters and, subsequently, the optimal partition. Moreover, let us remark that a value of $M = 3$ is likely to overfit the model by splitting the data in a too large number of clusters. This is evident in Figure 4.10, in Figure 4.13 and in Figure 4.16. By visual inspection, it is clear that 3 or 4 clusters represent well the data. Additional clusters include only few data and therefore are redundant.

Let us analyse the behaviour of the data in the most representative clusters. In Figure 4.9 one can see that the yellow cluster contains data with few recurrent events, which are characterised by an increasing trend. In the orange cluster, longer trajectories are included, and their similar pattern is evident by the lowest peak at the third gap time. In the red cluster, an oscillating behaviour is observable. As far as the violet cluster is concerned, recurrent events with a lowest peak at the second gap time are grouped. These patterns can be found also in Figure 4.9 and in Figure 4.15 (apart from the label switching that causes a colour change), thus confirming that the optimal partition is robust with respect to the prior specification of the base measure G_0.

In order to be even more general with respect to the DP prior specification, one could introduce a prior distribution for M. In the classical framework (see Escobar and West, 1995) a Gamma distribution for M is used, which leads to a conjugate model. If no prior information is available, this choice allows to specify a vague prior distribution that will further be localised by the data. Alternatively, one can exploit a prior belief on M and include it in the prior distribution for M.

All the other parameters, i.e. σ^2 and the regression coefficients $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J$ are robust with respect to the prior specification of the Dirichlet process and therefore their inferences are not reported here.

Conclusions and further developments

This work has proposed a new Bayesian semiparametric model to study recurrent event times. In particular, time-dependency among waiting times is taken into account through an autoregressive model, whose parameters are a sample from a Dirichlet process. Therefore, a clustering model on the items in the sample is induced by the model via the minimisation of a suitable loss function. In particular, clusters are created according to the entire trajectories of the event counts over the period of observation, i.e. observations are assumed to have the same number of recurrent events. Both fixed and time-dependent covariates may be included in this framework. Thus, this model can be useful for the management of health care services, whose interest is the prediction of the next hospitalisation in order to plan the resources appropriately.

A remarkable achievement is that this model is pretty robust with respect to the prior specification, which is a non-trivial issue in Bayesian nonparametrics. The clustering structure is, as expected, sensitive with respect to the total mass parameter of the DP prior. However, the choice of the other hyperparameters does not alter the results. Moreover, the Polya scheme adopted is, in the class of the Dirichlet Process Mixture models, very flexible because it allows to specify any kind of non-conjugate prior as the base measure of the Dirichlet Process. Given the complexity of the model, an efficient implementation was needed. Thus, the computational burden of this algorithm was reduced through the use of the Julia language, whose computational execution outperformed the one given by the R software.

As far as the drawbacks are concerned, this work needs a further generalisation in order to be completed. In fact, albeit already useful for hospital planning, the introduction of the eventual death of the patients would allow to use this model for medical purposes, too. In fact, until now we assumed that each patient experiences the same number of recurrent events before leaving the study. The joint modelling of the two processes, i.e.

gap times and survival, is the most intuitive continuation of this work. A survival model can be specified for the time-to-event (i.e. death) and an autoregressive model can be used for gap times.

Apart from the generalisation already discussed, other future developments are conceivable using this work as a starting point. For example, as a classical refinement of Bayesian non-parametric models, another level of hierarchy can be added in order to infer on the total mass parameter and to make the model less sensitive with respect to its choice. Moreover, the variance of the data density could be included in the DP sample. This latter choice would make the model more flexible, leading to heteroscedasticity in the groups of items.

Appendix A

Full conditionals and other calculations

A.1 Moments of the density of the data

The expected value of the observations $\boldsymbol{Y}_i$, conditionally to the parameters, is

$$\boldsymbol{\mu}_i = \begin{pmatrix} \mathbb{E}[Y_{i1}|\boldsymbol{x}_{i1}, \boldsymbol{\beta}_1, \boldsymbol{\alpha}_i, \sigma] \\ \mathbb{E}[Y_{i2}|\boldsymbol{x}_{i2}, \boldsymbol{\beta}_2, \boldsymbol{\alpha}_i, \sigma] \\ \vdots \\ \mathbb{E}[Y_{iJ}|\boldsymbol{x}_{iJ}, \boldsymbol{\beta}_J, \boldsymbol{\alpha}_i, \sigma] \end{pmatrix}$$

where

$$\mathbb{E}[Y_{i1}|param] = \boldsymbol{x}_{i1}^T\boldsymbol{\beta}_1 + \alpha_{i12}$$
$$\mathbb{E}[Y_{i2}|param] = \mathbb{E}\left[\mathbb{E}[Y_{i2}|Y_{i1}, param]|param\right] = \boldsymbol{x}_{i2}^T\boldsymbol{\beta}_2 + \alpha_{i21}(\boldsymbol{x}_{i1}^T\boldsymbol{\beta}_1 + \alpha_{i12}) + \alpha_{i22}.$$

The following recursive formula is then deduced:

$$\mathbb{E}[Y_{ij}|param] = \boldsymbol{x}_{ij}^T\boldsymbol{\beta}_j + \alpha_{ij1}\mathbb{E}[Y_{ij-1}|param] + \alpha_{ij2}.$$

As far as the covariance matrix Σ_i is concerned, conditionally to all the other parameters, we have

$$\Sigma_i = \begin{pmatrix} \mathrm{Var}(Y_{i1}) & \mathrm{Cov}(Y_{i1}, Y_{i2}) & \mathrm{Cov}(Y_{i1}, Y_{i3}) & \\ & \mathrm{Var}(Y_{i2}) & \mathrm{Cov}(Y_{i2}, Y_{i3}) & \\ & & \mathrm{Var}(Y_{i3}) & \\ & & & \ddots \end{pmatrix}.$$

We report here the calculation of the variances:

$$\begin{aligned}
\mathrm{Var}(Y_{i1}) &= \sigma^2 \\
\mathrm{Var}(Y_{i2}) &= \mathbb{E}[\mathrm{Var}(Y_{i2}|Y_{i1})] + \mathrm{Var}(\mathbb{E}[Y_{i2}|Y_{i1}]) \\
&= \sigma^2 + \mathrm{Var}(\alpha_{i21}Y_{i1} + \alpha_{i22}) = \sigma^2(1+\alpha_{i21}^2) \\
\mathrm{Var}(Y_{i3}) &= \dots \\
&= \sigma^2 + \alpha_{i31}^2\mathrm{Var}(Y_{i2}) = \sigma^2(1+\alpha_{i31}^2+\alpha_{i31}^2\alpha_{i21}^2).
\end{aligned}$$

The following recursive formula is then deduced:

$$\mathrm{Var}(Y_{ij}) = \sigma^2 + \alpha_{ij1}^2\mathrm{Var}(Y_{ij-1}).$$

One can also calculate the covariances

$$\begin{aligned}
\mathrm{Cov}(Y_{i1}, Y_{i2}) &= \mathbb{E}[Y_{i1}Y_{i2}] - \mathbb{E}[Y_{i1}]\mathbb{E}[Y_{i2}] = \mathbb{E}\left[Y_{i1}\mathbb{E}[Y_{i2}|Y_{i1}]\right] - \mathbb{E}[Y_{i1}]\mathbb{E}[Y_{i2}] \\
&= (\boldsymbol{x}_{i2}^T\boldsymbol{\beta}_2 + \alpha_{i22})\mathbb{E}[Y_{i1}] + \alpha_{i21}\mathbb{E}[Y_{i1}^2] - (\boldsymbol{x}_{i2}^T\boldsymbol{\beta}_2 + \alpha_{i22})\mathbb{E}[Y_{i1}] - \alpha_{i21}\left(\mathbb{E}[Y_{i1}^2]\right) \\
&= \alpha_{i21}\sigma^2
\end{aligned}$$

where we exploited the fact that

$$\mathbb{E}[Y_{i1}^2] = \mathrm{Var}(Y_{i1}) + \left(\mathbb{E}[Y_{i1}]\right)^2.$$

Analogously, one can write the recursive formula

$$\mathrm{Cov}(Y_{i1}, Y_{ij}) = \alpha_{ij1}\alpha_{ij-11}\dots\alpha_{i21}\sigma^2.$$

Therefore, the covariance matrix is filled with the following values

$$\Sigma_i = \sigma^2 \begin{pmatrix}
1 & \alpha_{i21} & \alpha_{i31}\alpha_{i21} & \alpha_{i41}\alpha_{i31}\alpha_{i21} & \\
 & 1+\alpha_{i21}^2 & \alpha_{i31}(1+\alpha_{i21}^2) & \alpha_{i41}\alpha_{i31}(1+\alpha_{i21}^2) & \\
 & & 1+\alpha_{i31}^2+\alpha_{i31}^2\alpha_{i21}^2 & \alpha_{i41}(1+\alpha_{i31}^2+\alpha_{i31}^2\alpha_{i21}^2) & \\
 & & & & \ddots
\end{pmatrix}.$$

A.2 Full conditional for σ^2

Given the model

$$\begin{aligned}
&\boldsymbol{Y}_i|\boldsymbol{\alpha}_i, \boldsymbol{x}_i, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_{n_i}, \sigma \overset{\text{ind}}{\sim} \mathcal{N}_{n_i}(\boldsymbol{\mu}_i, \Sigma_i) \\
&\sigma^2 \sim \text{inv-gamma}\left(\frac{\nu_0}{2}, \frac{\nu_0\sigma_0^2}{2}\right),
\end{aligned}$$

we find a simple expression for the posterior distribution. In fact, recalling the alternative parametrisation of the density of the data $k(\boldsymbol{y}; \boldsymbol{\alpha}_i, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{n_i}, \boldsymbol{x}_i, \sigma)$ as a function of the previous gap times, one can write

$$\begin{aligned}
\mathcal{L}(\sigma^2|\mathbb{Y}, rest) &\propto L(\mathbb{Y}; \sigma^2, rest)\mathcal{L}(\sigma^2) \\
&= \prod_{i=1}^{n} \left[\left(\frac{1}{(2\pi)^{n_i} \det(\sigma^2 \mathbb{I}_{n_i})} \right)^{\frac{1}{2}} e^{-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{\mu}_i)^T (\sigma^2 \mathbb{I}_{n_i})^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_i)} \right] \\
&\qquad \cdot \frac{\left(\frac{\nu_0 \sigma_0^2}{2}\right)^{\frac{\nu_0}{2}}}{\Gamma\left(\frac{\nu_0}{2}\right)} \left(\frac{1}{\sigma^2}\right)^{\frac{\nu_0}{2}+1} e^{-\frac{\nu_0 \sigma_0^2}{2}\frac{1}{\sigma^2}} \\
&\propto \prod_{i=1}^{n} \left[\left(\frac{1}{\sigma^2}\right)^{\frac{n_i}{2}} e^{-\frac{1}{2\sigma^2}(\boldsymbol{y}_i - \boldsymbol{\mu}_i)^T (\boldsymbol{y}_i - \boldsymbol{\mu}_i)} \right] \\
&\qquad \cdot \left(\frac{1}{\sigma^2}\right)^{\frac{\nu_0}{2}+1} e^{-\frac{\nu_0 \sigma_0^2}{2}\frac{1}{\sigma^2}} \\
&\propto \left(\frac{1}{\sigma^2}\right)^{\frac{\nu_0 + \sum_{i=1}^n n_i}{2}+1} e^{-\frac{1}{2\sigma^2}\left(\nu_0 \sigma_0^2 + \sum_{i=1}^n (\boldsymbol{y}_i - \boldsymbol{\mu}_i)^T (\boldsymbol{y}_i - \boldsymbol{\mu}_i)\right)}.
\end{aligned}$$

Hence we have

$$\sigma^2|\mathbb{Y}, rest \sim \text{inv-gamma}\left(\frac{\nu_0 + \sum_{i=1}^n n_i}{2}, \frac{\nu_0 \sigma_0^2 + \sum_{i=1}^n (\boldsymbol{y}_i - \boldsymbol{\mu}_i)^T (\boldsymbol{y}_i - \boldsymbol{\mu}_i)}{2} \right),$$

which, in the case of trajectories of length J, i.e. when $\sum_{i=1}^n n_i = nJ$, is the same expression of (3.10).

A.3 Full conditionals for $\boldsymbol{\beta}_j$

Given the model

$$\begin{aligned}
&\boldsymbol{Y}_i|\boldsymbol{\alpha}_i, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J, \boldsymbol{x}_i, \sigma \overset{\text{ind}}{\sim} \mathcal{N}_J(\boldsymbol{\mu}_i, \Sigma_i) \\
&\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J \overset{\text{iid}}{\sim} \mathcal{N}_p(\mathbf{0}, \beta_0^2 \cdot \mathbb{I}_p).
\end{aligned}$$

we want here to write the full conditionals for the update of $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J$.

For each observation, let us define the transformed data

$$\begin{aligned}
&\widetilde{Y}_{i1} = Y_{i1} - \alpha_{i11} \\
&\widetilde{Y}_{i2} = Y_{i2} - \alpha_{i21} Y_{i1} - \alpha_{i22} \\
&\vdots \\
&\widetilde{Y}_{iJ} = Y_{in_i} - \alpha_{in_i 1} Y_{iJ-1} - \alpha_{iJ2}
\end{aligned}$$

and the vectorial quantities

$$\begin{aligned}
&\widetilde{\boldsymbol{Y}}_1 = (\widetilde{Y}_{11}, \widetilde{Y}_{21}, \dots, \widetilde{Y}_{n1}) \\
&\vdots \\
&\widetilde{\boldsymbol{Y}}_j = (\widetilde{Y}_{1j}, \widetilde{Y}_{2j}, \dots, \widetilde{Y}_{nj}) \quad \forall j = 2, \dots, J.
\end{aligned}$$

Therefore we can update the $\boldsymbol{\beta}_j$'s one at the time. In fact, for each gap time the framework is the following (we write here the first gap time, but the full conditionals are the same $\forall j = 1, \dots, J$):

$$\begin{cases}
\widetilde{\boldsymbol{Y}}_1 | \mathbb{X}, \boldsymbol{\beta}_1, \sigma, \boldsymbol{\alpha} \sim \mathcal{N}_n(\mathbb{X}\boldsymbol{\beta}_1, \sigma^2 \mathbb{I}_n) \\
\boldsymbol{\beta}_1 \sim \mathcal{N}_p(\mathbf{0}, \beta_0^2 \cdot \mathbb{I}_p).
\end{cases}$$

The posterior density is then

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\beta}_1 | \tilde{\boldsymbol{Y}}_1, rest) &\propto L(\tilde{\boldsymbol{Y}}_1; \boldsymbol{\beta}_1, rest) \mathcal{L}(\boldsymbol{\beta}_1) \\
&= \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{n}{2}} e^{-\frac{1}{2\sigma^2}(\tilde{\boldsymbol{y}}_1 - \mathbb{X}\boldsymbol{\beta}_1)^T (\tilde{\boldsymbol{y}}_1 - \mathbb{X}\boldsymbol{\beta}_1)} \left(\frac{1}{2\pi\beta_0^2} \right)^{\frac{p}{2}} e^{-\frac{1}{2\beta_0^2} \boldsymbol{\beta}_1^T \mathbb{I}_p \boldsymbol{\beta}_1}.
\end{aligned} \tag{A.1}$$

Let us reparametrise the exponentiated quantity

$$\begin{aligned}
(\tilde{\boldsymbol{y}}_1 - \mathbb{X}\boldsymbol{\beta}_1)^T (\tilde{\boldsymbol{y}}_1 - \mathbb{X}\boldsymbol{\beta}_1) &= (\tilde{\boldsymbol{y}}_1^T - \boldsymbol{\beta}_1^T \mathbb{X}^T)(\tilde{\boldsymbol{y}}_1 - \mathbb{X}\boldsymbol{\beta}_1) \\
&= \tilde{\boldsymbol{y}}_1^T \tilde{\boldsymbol{y}}_1 - 2\boldsymbol{\beta}_1^T \mathbb{X}^T \tilde{\boldsymbol{y}}_1 + \boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_1
\end{aligned}$$

by adding and substituting the same quantity

$$S = (\tilde{\boldsymbol{y}}_1 - \mathbb{X}\hat{\boldsymbol{\beta}}_1)^T (\tilde{\boldsymbol{y}}_1 - \mathbb{X}\hat{\boldsymbol{\beta}}_1) = \tilde{\boldsymbol{y}}_1^T \tilde{\boldsymbol{y}}_1 - 2\hat{\boldsymbol{\beta}}_1^T \mathbb{X}^T \tilde{\boldsymbol{y}}_1 + \hat{\boldsymbol{\beta}}_1^T \mathbb{X}^T \mathbb{X} \hat{\boldsymbol{\beta}}_1,$$

where

$$\hat{\boldsymbol{\beta}}_1 = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \tilde{\boldsymbol{y}}_1.$$

We then get

$$\begin{aligned}
(\tilde{\boldsymbol{y}}_1 - \mathbb{X}\boldsymbol{\beta}_1)^T (\tilde{\boldsymbol{y}}_1 - \mathbb{X}\boldsymbol{\beta}_1) &= \boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_1 + 2\hat{\boldsymbol{\beta}}_1^T \mathbb{X}^T \tilde{\boldsymbol{y}}_1 - 2\boldsymbol{\beta}_1^T \mathbb{X}^T \tilde{\boldsymbol{y}}_1 - \hat{\boldsymbol{\beta}}_1^T \mathbb{X}^T \mathbb{X} \hat{\boldsymbol{\beta}}_1 + S \\
&= \boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_1 + 2(\hat{\boldsymbol{\beta}}_1^T - \boldsymbol{\beta}_1^T) \mathbb{X}^T \mathbb{X} (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \tilde{\boldsymbol{y}}_1 - \hat{\boldsymbol{\beta}}_1^T \mathbb{X}^T \mathbb{X} \hat{\boldsymbol{\beta}}_1 + S \\
&= \boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_1 + 2\hat{\boldsymbol{\beta}}_1^T \mathbb{X}^T \mathbb{X} \hat{\boldsymbol{\beta}}_1 - 2\boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \hat{\boldsymbol{\beta}}_1 - \hat{\boldsymbol{\beta}}_1^T \mathbb{X}^T \mathbb{X} \hat{\boldsymbol{\beta}}_1 + S \\
&= \boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_1 + \hat{\boldsymbol{\beta}}_1^T \mathbb{X}^T \mathbb{X} \hat{\boldsymbol{\beta}}_1 - 2\boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \hat{\boldsymbol{\beta}}_1 + S \\
&= (\boldsymbol{\beta}_1 - \hat{\boldsymbol{\beta}}_1)^T \mathbb{X}^T \mathbb{X} (\boldsymbol{\beta}_1 - \hat{\boldsymbol{\beta}}_1) + S.
\end{aligned}$$

By plugging this quantity in (A.1) one obtains

$$\mathcal{L}(\boldsymbol{\beta}_1 | \tilde{\boldsymbol{Y}}_1, rest) \propto e^{-\frac{1}{2\sigma^2}(\boldsymbol{\beta}_1 - \hat{\boldsymbol{\beta}}_1)^T \mathbb{X}^T \mathbb{X} (\boldsymbol{\beta}_1 - \hat{\boldsymbol{\beta}}_1)} e^{-\frac{1}{2\beta_0^2} \boldsymbol{\beta}_1^T \mathbb{I}_p \boldsymbol{\beta}_1}.$$

Let us now rewrite the exponential in terms of a quadratic form in $\boldsymbol{\beta}_1$:

$$\begin{aligned}
&-\frac{1}{2\sigma^2}(\boldsymbol{\beta}_1-\hat{\boldsymbol{\beta}}_1)^T\mathbb{X}^T\mathbb{X}(\boldsymbol{\beta}_1-\hat{\boldsymbol{\beta}}_1)-\frac{1}{2\beta_0^2}\boldsymbol{\beta}_1^T\mathbb{I}_p\boldsymbol{\beta}_1\\
=&-\frac{1}{2}\left[(\boldsymbol{\beta}_1-\hat{\boldsymbol{\beta}}_1)^T\frac{\mathbb{X}^T\mathbb{X}}{\sigma^2}(\boldsymbol{\beta}_1-\hat{\boldsymbol{\beta}}_1)+\boldsymbol{\beta}_1^T\frac{\mathbb{I}_p}{\beta_0^2}\boldsymbol{\beta}_1\right]\\
=&-\frac{1}{2}\left[\boldsymbol{\beta}_1^T\frac{\mathbb{X}^T\mathbb{X}}{\sigma^2}\boldsymbol{\beta}_1-2\boldsymbol{\beta}_1^T\frac{\mathbb{X}^T\mathbb{X}}{\sigma^2}\hat{\boldsymbol{\beta}}_1+\hat{\boldsymbol{\beta}}_1^T\frac{\mathbb{X}^T\mathbb{X}}{\sigma^2}\hat{\boldsymbol{\beta}}_1+\boldsymbol{\beta}_1^T\frac{\mathbb{I}_p}{\beta_0^2}\boldsymbol{\beta}_1\right].
\end{aligned}$$

By defining the quantities

$$\begin{aligned}
B_n&=\left(\frac{\mathbb{X}^T\mathbb{X}}{\sigma^2}+\frac{\mathbb{I}_p}{\beta_0^2}\right)^{-1},\\
\boldsymbol{b}_{1n}&=\left(\frac{\mathbb{X}^T\mathbb{X}}{\sigma^2}+\frac{\mathbb{I}_p}{\beta_0^2}\right)^{-1}\left(\frac{\mathbb{X}^T\mathbb{X}\hat{\boldsymbol{\beta}}_1}{\sigma^2}\right)=B_n\frac{\mathbb{X}^T\widetilde{\boldsymbol{y}}_i}{\sigma^2},
\end{aligned}$$

and noticing that

$$B_n^{-1}\boldsymbol{b}_{1n}=\frac{\mathbb{X}^T\mathbb{X}\hat{\boldsymbol{\beta}}_1}{\sigma^2}=\frac{\mathbb{X}^T\widetilde{\boldsymbol{y}}_i}{\sigma^2},$$

we rewrite

$$\begin{aligned}
&-\frac{1}{2\sigma^2}(\boldsymbol{\beta}_1-\hat{\boldsymbol{\beta}}_1)^T\mathbb{X}^T\mathbb{X}(\boldsymbol{\beta}_1-\hat{\boldsymbol{\beta}}_1)-\frac{1}{2\beta_0^2}\boldsymbol{\beta}_1^T\mathbb{I}_p\boldsymbol{\beta}_1\\
=&-\frac{1}{2\sigma^2}\left[\boldsymbol{\beta}_1^TB_n^{-1}\boldsymbol{\beta}_1-2\boldsymbol{\beta}_1^TB_n^{-1}\boldsymbol{b}_{1n}+\hat{\boldsymbol{\beta}}_1^TB_n^{-1}\boldsymbol{b}_{1n}\right]\\
=&-\frac{1}{2\sigma^2}\left[\boldsymbol{\beta}_1^TB_n^{-1}\boldsymbol{\beta}_1-2\boldsymbol{\beta}_1^TB_n^{-1}\boldsymbol{b}_{1n}+\boldsymbol{b}_{1n}^TB_n^{-1}\boldsymbol{b}_{1n}-\boldsymbol{b}_{1n}^TB_n^{-1}\boldsymbol{b}_{1n}+\hat{\boldsymbol{\beta}}_1^TB_n^{-1}\boldsymbol{b}_{1n}\right]\\
=&-\frac{1}{2\sigma^2}\left[(\boldsymbol{\beta}_1-\boldsymbol{b}_{1n})^TB_n^{-1}(\boldsymbol{\beta}_1-\boldsymbol{b}_{1n})^T-\boldsymbol{b}_{1n}^TB_n^{-1}\boldsymbol{b}_{1n}+\hat{\boldsymbol{\beta}}_1^TB_n^{-1}\boldsymbol{b}_{1n}\right].
\end{aligned}$$

Therefore the posterior density is

$$\mathcal{L}(\boldsymbol{\beta}_1|\tilde{\boldsymbol{Y}}_1,rest)\propto e^{-\frac{1}{2\sigma^2}(\boldsymbol{\beta}_1-\boldsymbol{b}_{1n})^TB_n^{-1}(\boldsymbol{\beta}_1-\boldsymbol{b}_{1n})^T},$$

and a Gibbs sampling scheme can be adopted using

$$\boldsymbol{\beta}_1|\widetilde{\boldsymbol{Y}}_1,rest\sim\mathcal{N}_p(\boldsymbol{b}_{1n},B_n).$$

The same strategy can be adopted for each of the successive gap times

$$\begin{cases}\widetilde{\boldsymbol{Y}}_j|\boldsymbol{Y}_{j-1},\mathbb{X},\boldsymbol{\beta}_j,\sigma,\boldsymbol{\alpha}\sim\mathcal{N}_n(\mathbb{X}\boldsymbol{\beta}_j,\sigma^2\mathbb{I}_n)\\ \boldsymbol{\beta}_j\sim\mathcal{N}_p(\mathbf{0},\beta_0^2\mathbb{I}_p)\end{cases}\Rightarrow\boldsymbol{\beta}_j|\widetilde{\boldsymbol{Y}}_j,rest\sim\mathcal{N}_p(\boldsymbol{b}_{jn},B_n)\quad\forall j=2,\dots,J.$$

Appendix B

Implementation in Julia

In this appendix, the implementation in Julia language is presented.

First of all, a function that calculates the mean vector and the covariance of the density kernel $k(\boldsymbol{y}; \boldsymbol{\alpha}_i, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J, \boldsymbol{x}_i, \sigma)$, given the data, is needed. Straightforward it is implementable another function that calculates the density of the J-variate normal, given the data and the parameters.

```
# Function that returns the mean vector and the variance matrix of the density of the data
function moments(y::Vector, x::Array{Float64,2}, beta::Array{Float64,2}, alpha::Array{Float64,2}, sig::Float64)

   J = convert(Int64, (length(alpha)+1)/2)
   media = zeros(J)
   diagonal = zeros(J)
   Sigma = zeros(Float64, J, J)

   media[1] = dot(vec(x[1,:]),beta[:,1]) + alpha[1]
   diagonal[1] = sig^2

   if J > 1
      for j in 2:J
         media[j] = dot(vec(x[j,:]),beta[:,j]) + alpha[2*j-2]*y[j-1] + alpha[2*j-1]
         diagonal[j] = sig^2
      end
   end
   Sigma = diagm(diagonal)

   return media, Sigma
end;

# Function that returns the density of the J-variate normal with given parameters, in a specific point R^J
function f_dens(y::Vector, x::Array{Float64,2}, beta::Array{Float64,2}, alpha::Array{Float64,2}, sig::Float64)
```

```
25
26    J = convert(Int64, (length(alpha)+1)/2)
27
28    if (length(y) != J)
29       print("Dimension of Y should be equal to J")
30       return -1
31    else
32       media, Sigma = moments(y, x, beta, alpha, sig)
33    end
34
35    if J == 1
36       d1 = Normal(media[], sqrt(Sigma)[])
37       return pdf(d1, y)
38    else
39       d1 = MvNormal(media, Sigma)
40       return pdf(d1, y)
41    end
42 end;
```

Listing B.1: Functions used to calculate the density of the data, given a set of parameters.

Then, the part of Gibbs Sampling relative to the label update is written, according to Algorithm 8 in Neal (2000).

```
1  # This function is the implementation of Algorithm 8 in Neal (2000).
2  # The input s is the configuration vector (s_i takes values in 1, ..., k, where k is
       the number of rows of phi, i.e. the number of clusters).
3  # s_i is the label of the ith observation (if s_i = s_j then the observations are in
       the same cluster).
4  # phi is the vector of the unique cluster specific parameters
5  # nphi denotes the size of each cluster
6  # y is the data matrix (dimension n*J)
7  # a is the total mass of the DP
8  function samp_conf(iter::Int, s::Array{Int,2}, phi::Array{Float64,2}, sig::Array{
       Float64,1}, nphi::Array{Int,1}, y::Array{Float64,2}, x::Array{Float64,2}, ni::
       Array{Int,1}, beta::Array{Float64,3}, a::Float64, m::Int)
9
10    n, J = size(y)
11    k = length(nphi)
12    ncov = size(x)[2]
13
14    d1 = Uniform(-1, 1)
15    d2 = Normal(0,10)
16
17    s_act = copy(s[iter-1,:])
18    phi_act = copy(phi)
19    beta_act = copy(reshape(beta[iter-1,:,:], ncov, J))
20
21    for i in 1:n
22       # actual covariate indices
23       idx_cov = copy(collect(((i-1)*J+1):(J*i)))
24
25       if nphi[s_act[i]]==1 # case when s_i is the only observation in a cluster
```

```
        k -= 1
        deleteat!(nphi, s_act[i])
        app = phi_act[s_act[i],:]
        phi_act = phi_act[1:size(phi_act,1) .!= s_act[i],:]

        # now the elements of s vary in 1, ..., k- (which is k-1)
        ind = s_act .> s_act[i]
        s_act[ind] -= 1

        phiaug = zeros(m-1, 2*J-1)
        for j in 1:(2*J-1)
           if j%2 == 1
              phiaug[:,j] = copy(rand(d2, m-1))
           else
              phiaug[:,j] = copy(rand(d1, m-1))
           end
        end
        phiaug = vcat(app, phiaug) # augmenting vector

        probold = Array(Float64, k)
        for h in 1:k
           probold[h] = log(nphi[h]) + log(f_dens(vec(y[i,:]), x[idx_cov,:], beta_act,
                phi_act[h,:], sig[iter-1]))
        end
        probnew = Array(Float64, m)
        for h in 1:m
           probnew[h] = log(a) - log(m) + log(f_dens(vec(y[i,:]), x[idx_cov,:], beta_
               act, phiaug[h,:], sig[iter-1]))
        end

        # normalisation trick with log-sum-exp
        log_probs = copy(cat(1, probold, probnew))
        norm_probs = copy(exp(log_probs - log(sum(exp(log_probs)))))

        if isprobvec(vec(norm_probs)) # check for underflow
           s_act[i] = rand(Categorical(vec(norm_probs)), 1)[]
        else
           print("WARNING: underflow \n")
        end

        if s_act[i]<=k # the new configuration is in one of the old clusters
           nphi[s_act[i]] += 1
        else # new cluster
           phi_act = vcat(phi_act,phiaug[s_act[i]-k,:])
           push!(nphi, 1)
           s_act[i] = k+1
           k += 1
        end

     else # case when nphi[s[i]]>1
        nphi[s_act[i]] -= 1

        phiaug = zeros(m, 2*J-1)
```

```
        for j in 1:(2*J-1)
           if j%2 == 1
              phiaug[:,j] = copy(rand(d2, m))
           else
              phiaug[:,j] = copy(rand(d1, m))
           end
        end

        probold = Array(Float64, k)
        for h in 1:k
           probold[h] = log(nphi[h]) + log(f_dens(vec(y[i,:]), x[idx_cov,:], beta_act,
                phi_act[h,:], sig[iter-1]))
        end
        probnew = Array(Float64, m)
        for h in 1:m
           probnew[h] = log(a) - log(m) + log(f_dens(vec(y[i,:]), x[idx_cov,:], beta_
               act, phiaug[h,:], sig[iter-1]))
        end

        # normalisation trick with log-sum-exp
        log_probs = copy(cat(1, probold, probnew))
        norm_probs = copy(exp(log_probs - log(sum(exp(log_probs)))))

        if isprobvec(vec(norm_probs)) # check for underflow
           s_act[i] = rand(Categorical(vec(norm_probs)), 1)[]
        else
           print("WARNING: underflow \n")
        end

        if s_act[i]<=k # the new configuration is in one of the old clusters
           nphi[s_act[i]] += 1
        else # new cluster
           phi_act = vcat(phi_act,phiaug[s_act[i]-k,:])
           push!(nphi, 1)
           s_act[i] = k+1
           k = k+1
        end
     end
  end

  s[iter,:] = copy(s_act);

  return(phi_act)
end;
```

Listing B.2: Implementation of Algorithm 8 in Neal (2000)

After the update of the labels, the cluster specific parameters $\boldsymbol{\alpha}^*$ have to be updated with a step of Metropolis-within-Gibbs algorithm as in (3.8).

```
# Metropolis step for the update of the cluster specific parameters alpha*.
function block_upd(iter::Int, s::Array{Int, 2}, phi::Array{Float64,2}, sig::Array{
    Float64,1}, nphi::Array{Int,1}, x::Array{Float64,2}, ni::Array{Int,1}, beta::Array
```

```
 {Float64,3}, y::Array{Float64,2})

n = size(s)[2]
k = length(nphi)
J = size(y)[2]
ncov = size(x)[2]

# Let us define the proposal distribution
sig_prop = zeros(2*J-1)
for i = 1:(2*J-1)
   if i%2 ==1
      sig_prop[i] = 0.01
   else
      sig_prop[i] = 0.001
   end
end
phi_act = copy(phi)
beta_act = copy(reshape(beta[iter-1,:,:], ncov, J))

d1 = Uniform(-1, 1)
d2 = Normal(0,10)
sig2_alpha = 100
mu_alpha = 0

for h in 1:(2*J-1)
   for i in 1:k
      # data indices of the cluster i
      idx = findin((vec(s[iter,:]) .== i), true)
      # data rescaling
      if h%2 == 1 # Gibbs sampler step
         time = copy(convert(Int64, floor((h+2)/2)))
         Y_tilde = zeros(nphi[i])
         cont = 1
         for j in idx
            idx_cov = copy(collect(((j-1)*J+1):(J*j)))
            if h == 1
               Y_tilde[cont] = copy(y[j,time] - dot(vec(x[idx_cov[time],:]), beta_
                   act[:,time]))
            else
               Y_tilde[cont] = copy(y[j,time]-dot(vec(x[idx_cov[time],:]), beta_act
                   [:,time])-y[j,time-1]*phi_act[i,h-1])
            end
            cont += 1
         end

         # moments of the full conditional
         mu_phi = copy((sig2_alpha*sum(Y_tilde)+mu_alpha*sig[iter-1]^2)/(sig2_alpha*
             nphi[i]+sig[iter-1]^2))
         sig2_phi = copy((sig[iter-1]^2*sig2_alpha)/(sig2_alpha*nphi[i]+sig[iter
             -1]^2))

         phi_act[i,h] = copy(rand(Normal(mu_phi, sqrt(sig2_phi)), 1)[])

```

```
        else # Metropolis-within-Gibbs step
           # Propose a new value
           delta = copy(rand(TruncatedNormal(phi_act[i,h], sqrt(sig_prop[h]), -1, 1),
               1))
           phi_prop = copy(phi_act[i,:])
           phi_prop[h] = copy(delta[])

           log_ker = 0
           # log-prior
           log_ker = copy(logpdf(d1, delta) - logpdf(d1, phi_act[i,h]))

           for j in idx # update the log-likelihood
              idx_cov = copy(collect(((j-1)*J+1):(J*j)))
              log_ker += log(f_dens(vec(y[j,:]), x[idx_cov,:], beta_act, phi_prop, sig
                  [iter-1])) - log(f_dens(vec(y[j,:]), x[idx_cov,:], beta_act, phi_
                  act[i,:], sig[iter-1]))
           end

           # Evaluation of the acceptance rejection ratio
           log_ker = copy(min(0.0, log_ker))
           lgu = copy(log(rand(Uniform(0,1),1)))

           if (lgu .< log_ker)[] # the value delta is accepted and the new actual
               value of psi is delta
              phi_act[i,h] = copy(phi_prop[h])
           end
        end
     end
  end
  return (phi_act)
end;
```

Listing B.3: Metropolis-within-Gibbs for the update of the cluster-specific parameters.

We then need a Gibbs step in order to update the variance parameter σ, using the conjugate form we found in (3.10).

```
# Gibbs Sampling for the update of sigma.
function sigma_upd(iter::Int, s::Array{Int, 2}, phi::Array{Float64,2}, sig::Array{
    Float64,1}, nphi::Array{Int,1}, x::Array{Float64,2}, ni::Array{Int,1}, beta::Array
    {Float64,3}, y::Array{Float64,2})

  n = size(s)[2]
  k = length(nphi)
  J = size(y)[2]
  ncov = size(x)[2]

  beta_act = copy(reshape(beta[iter-1,:,:], ncov, J))

  nu0 = 4.02
  sigma0 = sqrt(2.02/4.02)

  my_sum = 0
```

```
   for i in 1:k
      idx = findin(s[iter,:], i)
      for j in idx
         idx_cov = copy(collect(((j-1)*J+1):(J*j)))
         m_i, sig_i = moments(vec(y[j,:]),x[idx_cov,:],beta_act,phi[i,:],sig[iter-1])
         my_sum += dot(vec(y[j,:]) - m_i, vec(y[j,:]) - m_i)
      end
   end

   prec = rand(Gamma((n*J + nu0)/2.0, 1.0/((nu0*sigma0^2 + somma)/2)), 1)[]
    sig[iter] = copy(1.0/sqrt(prec))
end;
```

Listing B.4: Full conditional sampling of σ.

As far as the update of the covariate parameters $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J$ are concerned, we need two functions. The first one allows us to rescale the data in order to write them as a classic linear model. The second one is the Gibbs Sampling of the posterior distribution of $\boldsymbol{\beta}$ for each time, as described in (3.11)

```
# Function that rescales the data from Y to Y_tilde so that we can write the full
    conditional of Beta with respect to the scaled data.
function scale_data(y::Array{Float64,2}, s::Array{Int,2}, phi::Array{Float64,2})

   n, J = size(y)
   k = size(phi)[1]
   y_hat = zeros(n, J)

   for h in 1:k
      idx = findin(s, h)
      y_hat[idx,1] = copy(y[idx,1] - phi[h,1])
      for j in 2:J
         y_hat[idx,j] = copy(y[idx,j] - phi[h,2*j-2]*y[idx,j-1] - phi[h,2*j-1])
      end
   end
   return(y_hat)
end;

# Gibbs sampling steps that allow us to update the Beta parameters one at the time.
function beta_upd(iter::Int, s::Array{Int,2}, phi::Array{Float64,2}, sig::Array{Float64
    ,1}, nphi::Array{Int,1}, beta::Array{Float64,3}, x::Array{Float64,2}, y::Array{
    Float64,2}, ni::Array{Int,1})

   n = size(s)[2]
   k = length(nphi)
   J = size(y)[2]
   ncov = size(x)[2]

   beta0 = 50.0

   y_hat = scale_data(y, s[iter,:], phi) # rescale the data Y to Y^tilde

```

```
   inter = copy(1:J:(n*J))
   for i in 1:J
      idx_cov = copy(inter + (i-1))
      Bn = inv(x[idx_cov,:]'*x[idx_cov,:]/(sig[iter]^2) + eye(ncov)/(beta0^2))
      bn = Bn*((x[idx_cov,:]'*y_hat[:,i])/(sig[iter]^2))
      beta[iter,:,i] = copy(rand(MvNormal(bn, Bn), 1))
   end
end;
```

Listing B.5: Functions used to sample from the full conditional of the covariate parameters $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J$.

Bibliography

Aalen, O., Borgan, O., and Gjessing, H. (2008). *Survival and event history analysis: a process point of view*. Springer Science, New York.

Antoniak, C. E. (1974). Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, **2**, 1152–1174.

Argiento, R., Guglielmi, A., and Pievatolo, A. (2009). A comparison of nonparametric priors in hierarchical mixture modelling for AFT regression. *Journal of Statistical Planning and Inference*, **139**, 3989–4005.

Argiento, R., Cremaschi, A., and Guglielmi, A. (2014). A "density-based" algorithm for cluster analysis using species sampling Gaussian mixture models. *Journal of Computational and Graphical Statistics*, **23**, 1126–1142.

Barcella, W., De Iorio, M., and Baio, G. (2015). Variable Selection for Covariate Dependent Dirichlet Process Mixtures of Regressions. *arXiv preprint arXiv:1508.00129*.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2014). Julia: A Fresh Approach to Numerical Computing. *arXiv preprint arXiv:1411.1607*.

Binder, D. A. (1978). Bayesian Cluster Analysis. *Biometrika*, **65**, 31–38.

Bush, C. A. and MacEachern, S. N. (1996). A semiparametric Bayesian model for randomised block designs. *Biometrika*, **83**, 275–285.

Chang, S.-H. and Wang, M.-C. (1999). Conditional regression analysis for recurrence time data. *Journal of the American Statistical Association*, **94**, 1221–1230.

Christensen, R., Johnson, W., Branscum, A., and Hanson, T. E. (2011). *Bayesian Ideas and Data Analysis: an introduction for scientists and statisticians*. CRC Press, Boca Raton.

Cook, R. J. and Lawless, J. F. (2007). *The statistical analysis of recurrent events*. Springer Science, New York.

Di Lucca, M. A., Guglielmi, A., Müller, P., Quintana, F. A., et al. (2013). A simple class of bayesian nonparametric autoregression models. *Bayesian Analysis*, **8**, 63–88.

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, **90**, 577–588.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, **1**, 209–230.

Ishwaran, H. and James, L. F. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, **96**, 161–173.

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

Jasra, A., Holmes, C., and Stephens, D. (2005). Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, **20**, 50–67.

Lau, J. W. and Green, P. J. (2007). Bayesian model-based clustering procedures. *Journal of Computational and Graphical Statistics*, **16**, 526–558.

MacEachern, S. N. (1999). "Dependent nonparametric processes". In: *ASA Proceedings of the Section on Bayesian Statistical Science*, 50–55.

MacEachern, S. N. and Müller, P. (1998). Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics*, **7**, 223–238.

Müller, P., Erkanli, A., and West, M. (1996). Bayesian curve fitting using multivariate normal mixtures. *Biometrika*, **83**, 67–79.

Müller, P., Quintana, F. A., Jara, A., and Hanson, T. (2015). *Bayesian nonparametric data analysis*. Springer Science, New York.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics*, **9**, 249–265.

Papaspiliopoulos, O. and Roberts, G. O. (2008). Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika*, **95**, 169–186.

Pitman, J. (1996). Some developments of the Blackwell-MacQueen urn scheme. *Statistics, Probability and Game Theory, IMS Lecture Notes Monograph Series*, **30**, 245–267.

Prentice, R. L., Williams, B. J., and Peterson, A. V. (1981). On the regression analysis of multivariate failure time data. *Biometrika*, **68**, 373–379.

Regazzini, E. (1996). *Impostazione non parametrica di problemi di inferenza statistica bayesiana*. Tech. rep. CNR-IMATI 96.21.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, **4**, 639–650.

Ringraziamenti

Molte persone mi hanno aiutato, in maniera più o meno consapevole, a realizzare questo lavoro di tesi e per questo motivo ci tengo a ringraziarle pur sapendo che, ad ogni nome menzionato, ne corrispondono altrettanti che rischiano di essere omessi.

Voglio innanzitutto ringraziare la principale artefice nonché relatrice di questa tesi, la professoressa Alessandra Guglielmi, senza i cui preziosi consigli e la cui fiducia questo lavoro non sarebbe mai stato possibile. Il secondo pensiero va alla mia famiglia. Grazie, papà, per avermi sempre incoraggiato a superare i miei limiti e per essermi stato di ispirazione per la tua visione moderna del mondo; grazie, Anna, per essere più che una sorella un'amica e un modello da seguire. Grazie a tutti gli zii, in particolare a Pieri e Daniela: il primo per essere stato sempre di conforto nei momenti di difficoltà, la seconda per avermi dato l'amore di una mamma.

Non posso omettere coloro i quali hanno vissuto con me il percorso universitario, condividendone ansie, paure e gioie. Ringrazio tutti gli Ing. Mat, compagni di studio e non, per essere un grande gruppo. Grazie, Diana, per esserci sempre stata con i tuoi saggi consigli. Grazie, Paola, per avermi saputo aspettare e per aver creduto nella nostra amicizia. Grazie, Jacopo, una spalla su cui piangere e sfogare (troppe) ansie, per il tuo affetto e per la tua capacità di comprensione anche quando ti portavo a Baghdad il sabato pomeriggio. Grazie, Anna, per "vivere in un mondo che non esiste", per le insalate da 15€ e per portare sempre l'allegria nelle nostre giornate. Grazie, Ludo e Tobi, per i giovedì sera, per le serate trash e per avermi subito incluso tra i vostri amici.

Non sarei chi sono oggi senza il mio gruppo di amici udinesi: Giovanni, Enes, Tommy, Toso, Ele, Simo e Rispo. A voi va il mio ringraziamento per aver formato il mio carattere durante il liceo, perché con voi per la prima volta mi sono sentito parte di un gruppo. Che sia sul divano del Sig. Terzi, a Trieste o a Forni, ritornare in Friuli mi è più dolce sapendo che ci siete voi. Grazie, infine, a tutti i miei amici dell'École Centrale: David, Jofre, Miguel, Gaia, Andreas, Joanna, Romain, Kike e Giorgia. Con voi ho vissuto forse l'esperienza più bella della mia vita e sono orgoglioso che ne facciate parte.