



Politecnico di Milano
M.Sc. in Mathematical Engineering
Bayesian statistics
AY 2020/2021

Bayesian Clustering of Functional Data

Ajroldi Niccolò*
Bortolotti Teresa†
Marchionni Edoardo‡

Abstract

This is a project report of *Bayesian Statistics* course, held by Professor A. Guglielmi, at Politecnico di Milano during academic year 2020/2021. The tutor of this work is R. Corradin, PhD.

Somatosensory Evoked Potential (SEP) are electrical activity occurring in the central nervous system following a stimulus, detected within 2 seconds after the stimulus over 1600 instants. SEP reflect the processing of information from the peripheral levels down to the cortical structures, and are perhaps useful in assessing the integrity of nerve conduction pathways and the different areas involved in the reception and processing of sensory stimuli. The patients under study, in a state of coma, underwent a neuropsychological evaluation within 72 hours after surgery. We consider a dataset composed of 26 multivariate functional observation with 4 components, each of which represents a Somatosensory Evoked Potential, detected within 2 seconds after the stimulus over 1600 time instants. The aim of the project is to implement a functional clustering algorithm in a Bayesian nonparametric framework in a univariate functional setting. In particular, we assume an infinite mixture model for our functional obervations with a Dirichlet Process as mixture distribution. Our aim is to sample from Dirichlet process in order to get a sample of the latent partition induced. A Gibbs sampler is implemented for our purposes. The algorithm and the model will be first tested on simulated data and then on clinical ones.

*Mathematical Engineering Student at Politecnico di Milano.

†Mathematical Engineering Student at Politecnico di Milano.

‡Mathematical Engineering Student at Politecnico di Milano.

Contents

1	Introduction	2
1.1	Dataset Presentation	2
1.2	Research Question	2
2	Mixture Model	3
2.1	Dimensionality Reduction	3
2.1.1	Covariance Operator	3
2.1.2	Mean Operator	3
2.2	Base Measure Specification	4
2.3	Hyperpriors	4
3	Sampling	4
3.1	Blocked Gibbs Sampler	5
3.2	Full Conditional Derivation	7
3.2.1	Full Conditional for Latent Parameters Z_j	7
3.2.2	Full Conditional for Hyperparameters Y_j	8
3.3	Sample in Logarithmic Scale	8
3.4	Parameters Elicitation	9
3.4.1	Hyperparameters	9
3.4.2	Mass	10
3.4.3	Prior for ϕ_t parameters	10
4	Simulated data	10
4.1	Uncorrelated Time Points	10
4.2	Exponential covariance function	11
4.2.1	Testing on GP with Uncorrelated Time Points	12
4.2.2	Testing on GP with Correlated Time Points	12
5	Clinical data	13
5.1	Data Preprocessing and Basis Expansion	13
5.2	Testing	14
5.2.1	Proposal of a New Model	16
5.3	Conclusions	16
A	Full Conditional for ϕ_t	19
B	Full Conditional for μ	19

1 Introduction

1.1 Dataset Presentation

Clinical data at our disposal come from a study conducted by two hospital units in Provincia di Treviso, Italy; in particular, from Unità Gravi Cerebrolesioni Acquisite of Ospedale Ca' Foncello in Treviso and Unità Gravi Cerebrolesioni e Miolesioni of Ospedale Riabilitativo di Alta Specializzazione in Motta di Livenza.

The cohort under study is composed of 26 patients, who underwent two stages of assessment of their levels of functionality: first a neurophysiological evaluation while in a coma status and under sedation, and then a recovery evaluation. During the first stage of the study, patients have been subjected to a stimulation of the median nerve through a needle electrode to the wrist. Somatosensory Evoked Potentials, which are the electrical modifications occurring in the central nervous system following the stimulus, are detected within 2 seconds after the stimulus. Detections of the evoked potential have been made in four different positions of the scalp, two frontal and two central, in order to measure its components: SLSEP - Short Latency Somatosensory Evoked Potential and PMLSEP - Pain-related Middle Latency Somatosensory Evoked Potential.

For each patient, we are hence given with four components of the evoked potential, which are PML and SL measured both on the left and on the right of the skull. We performed the analysis considering only one of these components, in particular the SL component measured by the electrode placed on the left lobe. This component is treated as a functional datum, evaluated in 1600 time points.

After leaving the coma state, patients underwent a period of rehabilitation, at the end of which their levels of recovery have been evaluated and considered with respect to three different scales:

- GOSE₀ - *Glasgow Outcome Scale Extended*, for the assessment of the functional outcome of the patient, with a focus on the independence of the patient in his everyday life. Low levels indicate a negative outcome, high levels indicate a positive outcome.
- LCF₀ - *Level of Cognitive Functioning*, for the assessment of the level of responsiveness of the subject, through a definition of the progression of the recovery. Low levels indicate a negative outcome, high levels indicate a positive outcome.
- DRS₀ - *Disability Rate Scale*, for the assessment of the aspects related to the handicap. Low levels indicate a positive outcome, high levels indicate a negative outcome.

Given the numerosity of patients in the study, indexes of the assessment of the outcome have been synthesized as follows:

- if GOSE₀ ≤ 4 then GOSE = 1, if GOSE₀ > 4 then GOSE = 2
- if LCF₀ ≤ 5 then LCF = 1, if LCF₀ > 5 then LCF = 2
- if DRS₀ ≥ 6 then DRS = 1, if DRS₀ < 6 then DRS = 2

The observed sample did not show any empirical evidence in favor of the hypothesis that index GOSE is associated to indexes DRS or LCF.

1.2 Research Question

The analysis aims at performing clustering of functional observations of Somatosensory Evoked Potential, and investigate the presence of patterns in the levels of recovery of patients, when they are assigned to different clusters.

2 Mixture Model

The main purpose is to cluster functional data at our disposal, tackling the problem in a univariate setting. In order to pursue this task, observed curves $x_1(t), \dots, x_n(t)$ are supposed to be realizations of random functions $X_1(t), \dots, X_n(t)$ over some suitable space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. We assume that the probability distribution of the observations is represented by a mixture model. Let $X_i \sim f$, then:

$$f(x) = \int_{\Theta} k(x, \theta) p(d\theta) \quad (1)$$

where probability p is a random probability measure that follows a Dirichlet Process and $\{k(\cdot, \theta)\}_{\theta}$ are kernels depending on some parameter $\theta \in \Theta$. Since Dirichlet Processes are almost sure discrete, the mixture can be reformulated as an infinite sum

$$f(x) = \sum_{i=1}^{\infty} k(x, \theta_i) p_i \quad (2)$$

where the weights p_i sum up to 1 almost surely.

In particular, we assume that the kernels of the mixture are Gaussian processes, where both the mean and the covariance operators play the role of the parameter θ . This leads us to recast our problem as follows:

$$\begin{aligned} X_i | \mu_i, R_i &\stackrel{ind}{\sim} GP(\mu_i, R_i) \\ (\mu_i, R_i) | G &\stackrel{iid}{\sim} G \\ G &\sim DP(\alpha, \tilde{H}) \end{aligned} \quad (3)$$

where μ_i is the mean function and R_i is the covariance operator of the Gaussian process.

2.1 Dimensionality Reduction

To handle the proposed model and to simulate from the above Dirichlet Process, we have to reduce the dimensionality of the latent random variables. For this purpose each functional observation is assumed to be the sum of two elements

$$X(t) = \mu(t) + \varepsilon \quad (4)$$

where $\mu(t)$ is a function representing a mean effect and $\varepsilon \sim GP(0, R)$ is a stochastic process representing a variance effect.

2.1.1 Covariance Operator

Regarding the covariance operator, we assume the following representation

$$R(t, t') = \begin{cases} \phi_t & \text{if } t = t' \\ 0 & \text{if } t \neq t' \end{cases} \quad (5)$$

2.1.2 Mean Operator

As far as the mean function is concerned, inspired by Scarpa and Dunson 2014, we resort to a representation on a basis system. Let $\{b_l(t)\}_{l=1, \dots, L}$ be that basis and assume it exactly spans the space \mathcal{X} . This allows us to represent any observation as

$$X(t) = \sum_{l=1}^L \beta_l b_l(t) \quad \forall t. \quad (6)$$

Defining $\beta = [\beta_1, \dots, \beta_L]^T$ and $\mathbf{b}(t) = [b_1(t), \dots, b_L(t)]^T$, we get the more compact representation

$$X(t) = \beta^T \mathbf{b}(t) \quad \forall t. \quad (7)$$

In this framework the mean operator μ of each observation can be expanded as follows

$$\mu(t) = \sum_{l=1}^L \mu_l b_l(t) \quad \forall t. \quad (8)$$

As above, defining the vector $\mu = [\mu_1, \dots, \mu_L]^T$ we can get the more compact representation

$$\mu(t) = \mu^T \mathbf{b}(t) \quad \forall t.. \quad (9)$$

From now on, we can refer to $\mu(t)$ and $X(t)$ by means of their random coefficients $\mu := [\mu_1, \dots, \mu_L]^T$ and $\beta := [\beta_1, \dots, \beta_L]^T$ respectively. Thanks to the above assumptions, we obtain the following simpler representation of the mixture model:

$$\begin{aligned} X_i | \mu_i, \{\phi_{it}\}_t &\stackrel{ind}{\sim} GP(\mu_i, R_i) \\ (\mu_i, \{\phi_{it}\}_t) | G &\stackrel{iid}{\sim} G \\ G &\sim DP(\alpha, H) \end{aligned} \quad (10)$$

2.2 Base Measure Specification

The Dirichlet Process is a prior over a probability measure. In order to characterize it, both the mass parameter and the base measure need to be specified. As far as the base measure H is concerned, H is the joint distribution of our latent random variables $(\mu, \{\phi_t\}_t) \sim H$ obtained marginalizing out G . We assume $\{\phi_t\}_t$ independent and identically distributed and $\forall t \mu \perp\!\!\!\perp \phi_t$. Finally, we can characterize H by specifying the marginal distributions of the latent parameters:

$$\mu \sim \mathcal{N}_L(\mathbf{m}_0, \Lambda_0) \quad (11)$$

$$\phi_t \stackrel{iid}{\sim} IG(c, d) \quad (12)$$

2.3 Hyperpriors

To get a more sophisticated model and to gain flexibility, we set a hyperprior structure on the parameters of the base measure for μ . We set a normal-inverse-Wishart distribution on the hyperparameters of the vector μ . In particular, we set $(\mathbf{m}_0, \Lambda_0) \sim NIW(\theta_0, k_0, \nu_0, \Delta_0)$, that has the following hierarchical structure:

$$\begin{aligned} \mu | \mathbf{m}_0, \Lambda_0 &\sim \mathcal{N}(\mathbf{m}_0, \Lambda_0) \\ \mathbf{m}_0 | \Lambda_0 &\sim \mathcal{N}\left(\theta_0, \frac{\Lambda_0}{k_0}\right) \\ \Lambda_0 &\sim IW(\nu_0, \Delta_0). \end{aligned}$$

3 Sampling

Our purpose is to sample from the joint posterior distribution of the random probability measure G and parameters. The goal is to perform inference on the latent partition of our observations and on the random density H . In order to do so, we will rely on a Gibbs' sampling algorithm that will be extensively described in the current section.

First, we assume a finite dimensional truncation of the Dirichlet process, with M kernels. This allows us to express the model in terms of a finite number of random variables and to sample from it through a blocked Gibbs sampler. In this framework, we augment the problem introducing the following *random variables*:

- $Z_j := (\mu_j, \sigma_j, \{\phi_{jt}\}_t)$, i.e. it denotes the j -th kernel-specific latent parameters
- K_i is the assignment categorical variable of our i -th datum, it takes values in $\{1, \dots, M\}$ and $K_i = j$ if observation i belongs to the j -th kernel
- $Y_j := (m_{0j}, \Lambda_{0j})$, i.e. it stands for the j -th kernel-specific latent hyperparameter
- p_j is the probability of belonging to the j -th kernel, it takes

Furthermore, for the simplicity of notation we introduce the following vectors:

- $\mathbf{X}_i := (X_i(t_0), \dots, X_i(T))^T$, where X_i is our i -th observation and t_0, \dots, T are the time instants at which we observed our data
- $\mathbf{K} := (K_1, \dots, K_n)$
- $\mathbf{Z} := (Z_1, \dots, Z_M)$
- $\mathbf{Y} := (Y_1, \dots, Y_M)$
- $\mathbf{p} := (p_1, \dots, p_M)$

It is noteworthy that the connection with the previous section follows by remarking that $(\mu_i, \{\phi_{it}\}_t) = Z_{K_i}$. In this context, our model can be rewritten as

$$\begin{aligned} X_i | \mathbf{Z}, \mathbf{K}, \mathbf{Y} &\stackrel{iid}{\sim} \text{Gaussian Process with parameters } Z_{K_i} \\ Z_j | \mathbf{Y} &\stackrel{iid}{\sim} H \quad Y_j \stackrel{iid}{\sim} \mathcal{L} \\ K_i | \mathbf{p} &\stackrel{iid}{\sim} \text{categorical}(M, \mathbf{p}) \end{aligned} \tag{13}$$

where $j = 1, \dots, M$, $i = 1, \dots, n$ and \mathbf{p} follows a truncated stick breaking construction that will be presented in the next subsection.

3.1 Blocked Gibbs Sampler

The method is a generalization of the algorithm described in Ishwaran and James 2001, it works by iteratively drawing values from the posterior distributions of the blocked variables:

$$\begin{aligned} &\mathbf{Y} | \mathbf{Z} \\ &\mathbf{Z} | \mathbf{K}, \mathbf{Y}, \mathbf{X}_1, \dots, \mathbf{X}_n \\ &\mathbf{K} | \mathbf{Z}, \mathbf{Y}, \mathbf{p}, \mathbf{X}_1, \dots, \mathbf{X}_n \\ &\mathbf{p} | \mathbf{K} \end{aligned} \tag{14}$$

Each draw $(\mathbf{Z}, \mathbf{K}, \mathbf{Y}, \mathbf{p})$ defines a random probability measure G_M :

$$G_M(\cdot) = \sum_{k=1}^M p_k \delta_{Z_k}(\cdot), \tag{15}$$

which provides a draw from the posterior $DP(\cdot | \alpha, H, X)$. On the other hand, the allocation variable \mathbf{K} provides a draw of the latent partition induced by the Dirichlet process.

The general idea of the algorithm is to assign at every iteration each observation to the most probable group, then draw the hyperparameters from the full conditional for each group and last draw parameters conditionally to the observations assigned to the group. As an initialization step, we allocate observations to different groups.

Now let K_1^*, \dots, K_m^* denote the set of current unique values of \mathbf{K} . The blocked Gibbs sampler works iterating the following steps:

1. Draw from the full conditional for \mathbf{Y}
2. Draw from the full conditional for \mathbf{Z}
3. Draw from the full conditional for \mathbf{K}
4. Draw from the full conditional for \mathbf{p}

Step 1: conditional for \mathbf{Z}

After initialization and after every following iteration, we are given with empty groups and groups to which at least one observation has been assigned. We distinguish the update rule for these two cases: parameters of an empty group are simulated from the original priors with hyperparameters simulated from the original hyperpriors, while parameters of a non-empty group are drawn from the full conditionals after having drawn the hyperparameters from their full conditionals and hence updated according to the observations assigned to the kernel.

Formally, let $\{K_1^*, \dots, K_m^*\}$ denote the set of r unique values of \mathbf{K} at the current iteration. For each cluster $j \in \{K_1^*, \dots, K_m^*\}^c$ (i.e. the empty clusters) simulate Y_j, Z_j :

$$\begin{aligned} Y_j &\sim \mathcal{L} \\ Z_j &\sim H(Z_j|Y_j) \end{aligned}$$

For each cluster $j \in \{K_1^*, \dots, K_m^*\}$ (i.e. the clusters to which at least one observation has been assigned) simulate $Y_j|Z_j$ and $Z_j|\mathbf{K}, Y_j, X_1, \dots, X_n$ as follows:

$$\begin{aligned} Y_j|Z_j &\sim f(Y_j|Z_j) \\ Z_j|\mathbf{K}, Y_j, X_1, \dots, X_n &\sim f(Z_j|\mathbf{K}, Y_j, X_1, \dots, X_n) \propto H(Z_j|Y_j) \prod_{\{i:K_i=j\}} f(X_i|Z_j) \end{aligned}$$

Step 2: conditional for \mathbf{K}

For each observation $i = 1, \dots, n$ and for every cluster j , evaluate the probability that the i -th observation belongs to cluster j as p_j times the kernel density evaluated in $(\mu_j, \phi_{j1}, \dots, \phi_{jT})$. The result is a M -dimensional vector of probabilities \mathbf{p}_i :

$$\mathbf{p}_i := (p_{1,i}, \dots, p_{M,i}) \propto (p_1 f(X_i|Z_1, Y_1), \dots, p_M f(X_i|Z_M, Y_M)) \quad (16)$$

In addition, simulate $K_i|\mathbf{Z}, \mathbf{Y}, \mathbf{p}, \mathbf{X}_1, \dots, \mathbf{X}_n \sim \text{categorical}(M, \mathbf{p}_i)$.

Step 3: conditional for \mathbf{p}

Update the weights for each group:

$$\begin{aligned}
p_1 &= V_1, \\
p_j &= V_j \prod_{l < j} (1 - V_l) , \text{ for } j = 2, \dots, M \\
\text{where } V_j &\stackrel{iid}{\sim} Beta \left(1 + M_j, \alpha + \sum_{l=k+1}^M M_l \right), \text{ for } j = 1, \dots, M-1 \\
V_M &= 1
\end{aligned} \tag{17}$$

where M_j is the number of observations belonging to cluster j , i.e. $M_j = \#\{i \in \{1, \dots, n\} : K_i = j\}$.

3.2 Full Conditional Derivation

In this section, we rely on two representations of the observations, which are equivalent, as we extensively explained in Section 2.1. Therefore, we will not further justify the usage of these representations here.

3.2.1 Full Conditional for Latent Parameters Z_j

The full conditional density of j -th kernel-specific parameters is of the form

$$f(Z_j | \mathbf{K}, Y_j, X_1, \dots, X_n) \propto H(Z_j | Y_j) \prod_{\{i:K_i=j\}} f(X_i | Z_j, Y_j) \tag{18}$$

Exploiting the vectors $\{\mathbf{X}_i\}_i$ and relying on the properties of Gaussian processes (i.e. multivariate Gaussian marginality), we have

$$f(\mathbf{X}_i | Z_j, Y_j) = \prod_{t=t_0}^T f(X_i(t) | Z_j, Y_j) = \prod_{t=t_0}^T \frac{1}{\sqrt{2\pi\phi_{jt}}} \exp \left\{ -\frac{(X_i(t) - \mu_j(t))^2}{2\phi_{jt}} \right\} \tag{19}$$

For the sake of notation convenience, from now on we drop the index j that denotes the kernel membership, keeping in mind that any full conditional is for the parameters of the j -th group.

Considering first the full conditional for ϕ_t and fixing $t \in \{t_0, \dots, T\}$, we have:

$$f(\phi_t | \text{rest}) \propto H(\phi_t) \prod_{\{i:K_i=j\}} f(X_i(t) | Z_j, Y_j). \tag{20}$$

Recall that the prior for ϕ_t is $\forall t \phi_t \stackrel{iid}{\sim} IG(c, d)$, the posterior distribution can easily derived:

$$\phi_t | \text{rest} \sim \text{inv-gamma} \left(c + \frac{r}{2}, d + \sum_{\{i:K_i=j\}} \frac{(X_i(t) - \mu(t))^2}{2} \right) \tag{21}$$

We refer to section A of the appendix for explicit computations.

We now aim to derive the full conditional for $\boldsymbol{\mu}$. Similarly to the case for ϕ_t we have:

$$f(\boldsymbol{\mu} | \text{rest}) \propto H(\boldsymbol{\mu} | Y_j) \prod_{\{i:K_i=j\}} f(\mathbf{X}_i | Z_j, Y_j)$$

For what concerns the first term, we recall that the prior for $\boldsymbol{\mu}$ is $\boldsymbol{\mu} \sim \mathcal{N}_L(\mathbf{m}_0, \Lambda_0)$. Therefore, the full conditional for $\boldsymbol{\mu}$ becomes:

$$\boldsymbol{\mu}|\text{rest} \sim \mathcal{N}_L(\mathbf{m}_r, \Lambda_r) \quad (22)$$

where

$$\begin{aligned} \mathbf{m}_r &= \Lambda_r \left(\Lambda_0^{-1} \mathbf{m}_0 + \left[\sum_{t=t_0}^T \frac{\mathbf{b}(t)\mathbf{b}(t)^T}{\phi_t} \right] \sum_{\{i:K_i=j\}} \boldsymbol{\beta}_i \right) \\ \Lambda_r &= \left(\Lambda_0^{-1} + r \left[\sum_{t=t_0}^T \frac{\mathbf{b}(t)\mathbf{b}(t)^T}{\phi_t} \right] \right)^{-1} \end{aligned}$$

We refer to section B of the appendix for detailed computations.

3.2.2 Full Conditional for Hyperparameters Y_j

The full conditional density of our j -th kernel-specific hyperparameters is of the form:

$$f(Y_j|Z_j) \propto H(Z_j|Y_j)\mathcal{L}(Y_j) \quad (23)$$

As above, for simplicity of notation, we drop the index j that denotes the kernel membership, keeping in mind that any full conditional is for the hyperparameters of the j -th group.

Finally, we have to derive full conditional for hyperparameters of $\boldsymbol{\mu}$. We recall that we set the following hyperprior structure $(\mathbf{m}_0, \Lambda_0) \sim \text{NIW}(\boldsymbol{\theta}_0, k_0, \nu_0, \Delta_0)$. Moreover, we recall that $H(\boldsymbol{\mu}|\mathbf{m}_0, \Lambda_0) \stackrel{\text{law}}{=} \mathcal{N}(\mathbf{m}_0, \Lambda_0)$. Hence we have that the full conditional law is:

$$\mathcal{L}(\mathbf{m}_0, \Lambda_0|\boldsymbol{\mu}) \propto \mathcal{N}(\boldsymbol{\mu}; \mathbf{m}_0, \Lambda_0) \times \text{NIW}(\mathbf{m}_0, \Lambda_0; \boldsymbol{\theta}_0, k_0, \nu_0, \Delta_0)$$

Since the normal-inverse-Wishart structure is conjugate with the normal distribution, we have that

$$\mathbf{m}_0, \Lambda_0|\boldsymbol{\mu} \sim \text{NIW}(\boldsymbol{\theta}_1, k_1, \nu_1, \Delta_1)$$

where

$$\begin{aligned} k_1 &= k_0 + 1 & \nu_1 &= \nu_0 + 1 \\ \boldsymbol{\theta}_1 &= \frac{k_0 \boldsymbol{\theta}_0 + \boldsymbol{\mu}}{k_1} \\ \Delta_1 &= \Delta_0 + \frac{k_0}{k_1} (\boldsymbol{\mu} - \boldsymbol{\theta}_0)(\boldsymbol{\mu} - \boldsymbol{\theta}_0)^T \end{aligned}$$

3.3 Sample in Logarithmic Scale

In Step 2 of the proposed Blocked Gibbs sampler, we have to sample from a categorical distribution. It is worth noticing that the probabilities of the categorical distribution are proportional to the likelihood (25). The evaluation of the likelihood in the different atoms of the different kernels requires the product of a high number factors. To better control this number, avoiding it attains values less than the machine ϵ , we compute this number in logarithmic scale. This means that, instead of (p_1, \dots, p_N) we have $(\log(p_1), \dots, \log(p_N))$. In order to obtain the probabilities and to sample, we apply the following transformation:

$$p_k = \frac{1}{\sum_{j=1}^N \exp(\log(p_j) - \log(p_k))} = \frac{\exp(\log(p_k))}{\sum_{j=1}^N \exp(\log(p_j))}, \quad (24)$$

It is noteworthy that the transformation works up to a translation on the logarithm scale, i.e defining $\log(p_j^*) = \log(p_j) + C$.

In our particular case, we have that for observation i ,

$$f(\mathbf{X}_i|Z_j, Y_j) = \prod_{t=t_0}^T \frac{1}{\sqrt{2\pi\phi_{jt}}} \exp \left\{ -\frac{(X_i(t) - \mu_j(t))^2}{2\phi_{jt}} \right\} \quad (25)$$

so that:

$$\log f(\mathbf{X}_i|Z_j, Y_j) = \sum_{t=t_0}^T \left[-\frac{1}{2} \log(2\pi\phi_{jt}) - \frac{(X_i(t) - \mu_j(t))^2}{2\phi_{jt}} \right]$$

Hence, we have

$$\begin{aligned} p_{j,i} &= p_j \cdot f(X_i|Z_j, Y_j) \\ \log p_{j,i} &= \log(p_j \cdot f(X_i|Z_j, Y_j)) = \log(p_j) + \sum_{t=t_0}^T \left[-\frac{1}{2} \log(2\pi\phi_{jt}) - \frac{(X_i(t) - \mu_j(t))^2}{2\phi_{jt}} \right]. \end{aligned}$$

3.4 Parameters Elicitation

The parameters to elicitate are those of the hyperpriors of $\boldsymbol{\mu}$, the mass α of the Dirichlet process and the parameters c and d of the inv-gamma prior of ϕ_t .

3.4.1 Hyperparameters

We proceed using our data for the elicitation of the parameters of the hyperprior. We recall we set a normal-inverse-Wishart structure on our data and referring to the notation of section 2.3 the parameters to elicitate are $\boldsymbol{\theta}_0$, κ_0 , ν_0 and Δ_0 .

First, we set the expected value of $\boldsymbol{\mu}_0$ equal to the empirical mean of the coefficients of the considered data with respect to the basis expansion, that is

$$\boldsymbol{\theta}_0 = \mathbb{E}[\mathbf{m}_0] = \frac{1}{n} \sum_{i=1}^n \boldsymbol{\beta}_i$$

Second, the expected value of Λ_0 is set equal to the empirical variance of the coefficients of the considered data with respect to the basis expansion, keeping ν_0 the lowest possible we get

$$\begin{aligned} \nu_0 &= L \\ \mathbb{E}[\Lambda_0] &= \frac{\Delta_0}{\nu_0 - L + 1} = \frac{1}{(n-1)} \sum_{i=1}^n (\boldsymbol{\beta}_i - \bar{\boldsymbol{\beta}})^2 \end{aligned}$$

Finally, κ_0 is selected equal to 0.1, in order to increase the variance of \mathbf{m}_0 , so to allow the algorithm to propose more flexible values of the coefficients of $\boldsymbol{\mu}$.

3.4.2 Mass

As far as the mass is concerned, at first we set it such that the number of expected clusters is equal to desired value; that is, for instance

$$\mathbb{E}[K_n] = \sum_{i=1}^n \frac{\alpha}{\alpha + j - 1} = 3 \iff \alpha \simeq 0.6$$

Since in such setting the algorithm performs very poorly in terms of mixing, allowing for much higher values of mass turns out to be very effective in obtaining a better mixing. Therefore, we test the algorithm with $\alpha \simeq 40$.

3.4.3 Prior for ϕ_t parameters

The setting of the parameters of the inv-gamma prior for ϕ_t has been particularly critical. We see that the most effective elicitation comes as result of a sort of trade-off between mixing performance of the algorithm and its ability to separate observations in clusters. Eventually, we see that the best performance is obtained by setting c and d such that

$$\mathbb{E}[\phi_t] = \frac{d}{c-1} = 1 \text{ and } \text{Var}(\phi_t) = \frac{d^2}{(c-1)^2(c-2)} = 0.2$$

which is reasonable, since we do not want ϕ_t to be so high that a curve could be assigned to a cluster that is not representative of it, and on the contrary we do not want ϕ_t to be so small that it becomes very difficult to propose a value of μ that generates a new cluster.

4 Simulated data

In order to assess performances of the specified model, we simulate data from three distinct Gaussian Processes, aiming at separating them in three clusters.

4.1 Uncorrelated Time Points

For each of the three groups, we generate $n = 10$ data $X_i(t)$, ($i = 1, \dots, n$ and $t = 1, \dots, T$) on a grid of $T = 100$ time points, according to $X_i(t)|\mu_i, \{\phi_{it}\}_t \stackrel{\text{ind}}{\sim} GP(\mu_i, R_i)$, specifying a sinusoidal mean function $\mu(t)$ and a covariance operator $R(t, t')$ coherent with our model: $R(t, t') = \begin{cases} \phi_t & \text{if } t = t' \\ 0 & \text{if } t \neq t' \end{cases}$, where we simulate ϕ_t from an inverse gamma distribution. In particular the mean operator of the three groups are sinus functions with

1. amplitude= 0.55, angular frequency= 0.2π , phase= 0
2. amplitude= 0.2, angular frequency= 0.2π , phase= 0.4
3. amplitude= 0.55, angular frequency= 0.35π , phase= 1.4

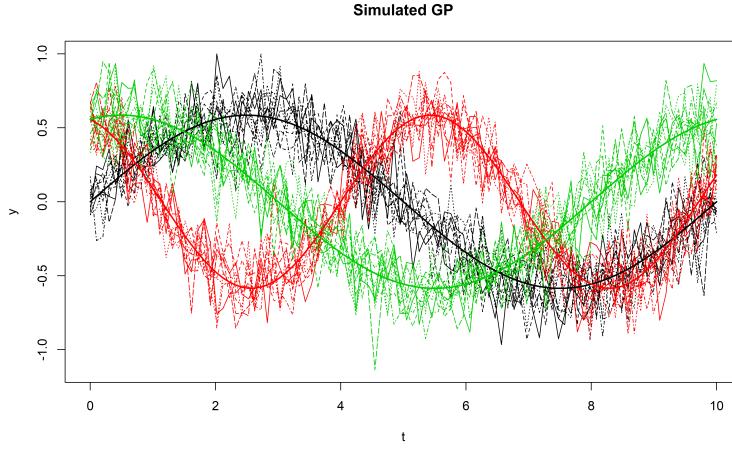


Figure 1: Simulated data with diagonal covariance matrix

In Figure 1 we report simulated data from the three Gaussian processes. In the framework of dimensionality reduction (see Section 2.1), we have to select an appropriate number of basis functions to represent our data. Since we are using periodic data, a natural choice is to truncate the basis system to 7 basis functions, that allows to capture 3 harmonic frequencies.

4.2 Exponential covariance function

It is noteworthy that the previous data construction does not violate the assumption of independence between time points. To test the performance of the model in a framework in which this assumption is not met, we simulate again data from three Gaussian processes, considering this time a discretization of an exponential covariance function of the form: $R(t, t') \propto e^{-\beta|t' - t|}$ over a 1D grid $[t_1, \dots, t_T]$, thus obtaining the PxP covariance matrix of values: $R_{i,j} = C(t_i, t_j) \propto e^{-\beta|t_i - t_j|}$.

In Figure 2 we report simulated data from the three Gaussian processes, with correlated times.

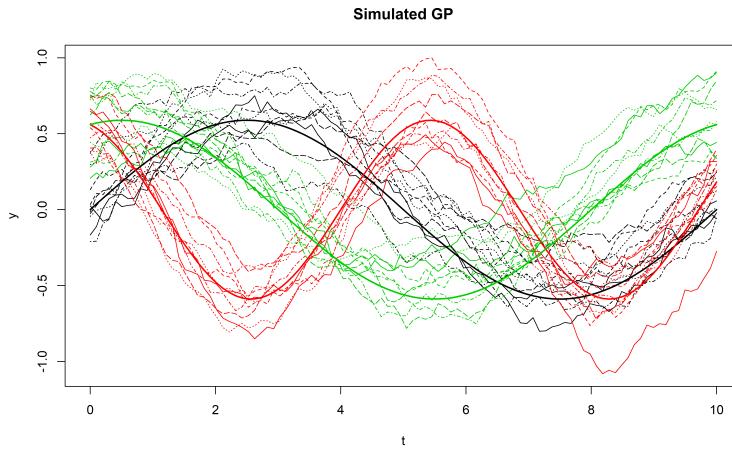


Figure 2: Simulated data with exponential covariance matrix

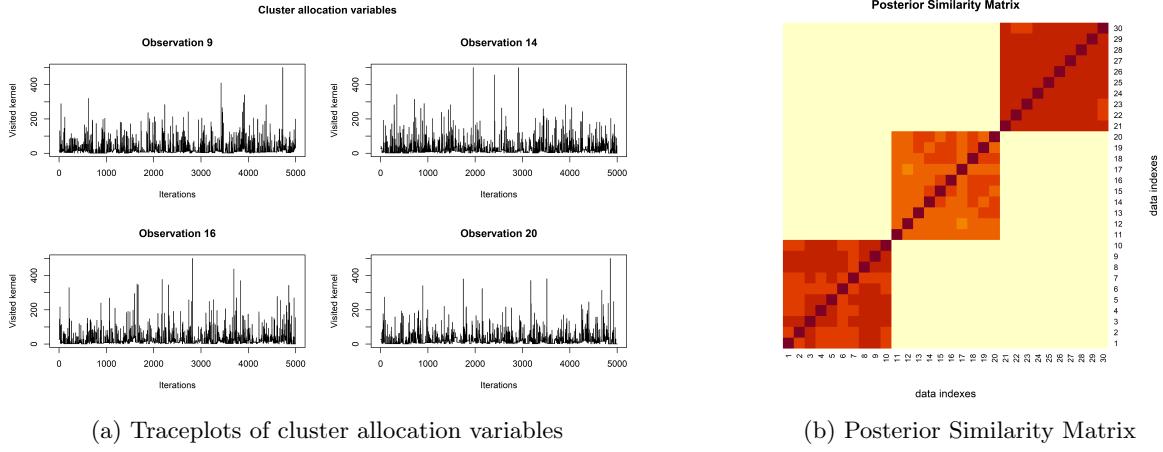


Figure 4: Traceplots and PSM for simulated data

4.2.1 Testing on GP with Uncorrelated Time Points

We ran posterior sampling for 5000 iterations after 10000 burnin iterations. Truncation level of stick breaking prior is fixed at 500. By minimizing Binder loss function, optimal allocation of observations separates data into three groups, leading to perfect clustering of simulated data. In [Figure 3b](#) we report posterior similarity matrix, while in [Figure 3a](#) we report traceplots of cluster allocation variables K_i for some observations i . We can appreciate good mixing and standard diagnostic tests like Geweke test showed no evidence against convergence of the chain.

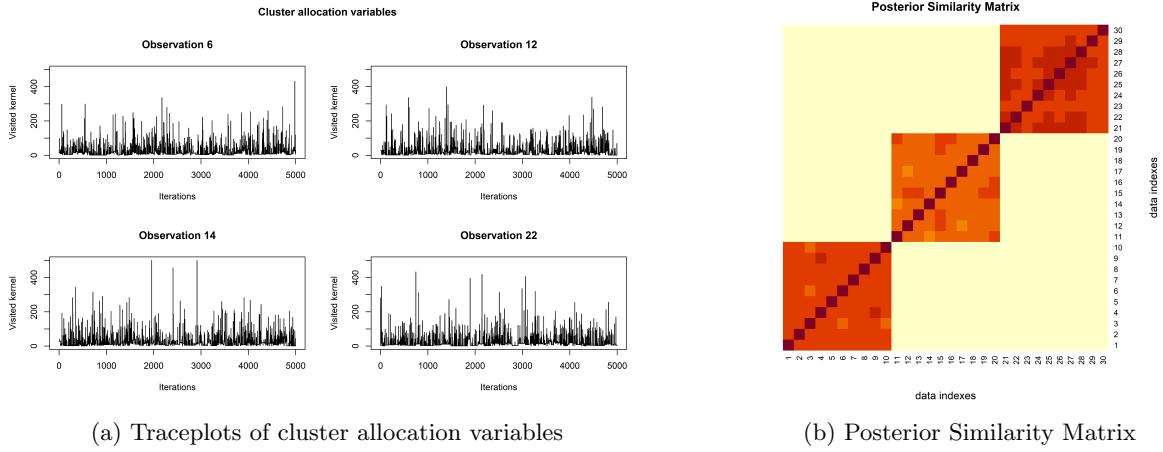


Figure 3: Traceplots and PSM for simulated data

4.2.2 Testing on GP with Correlated Time Points

Posterior inference is again performed for 5000 iterations after 10000 burnin iterations. Truncation level of stick breaking prior is fixed at 500. Despite the presence of a positive correlation within times, the algorithm is still able to propose suitable kernels and performances are not worsened. In [Figure 4b](#) we report the posterior similarity matrix, while in [Figure 4a](#) we report traceplots of cluster allocation variables.

5 Clinical data

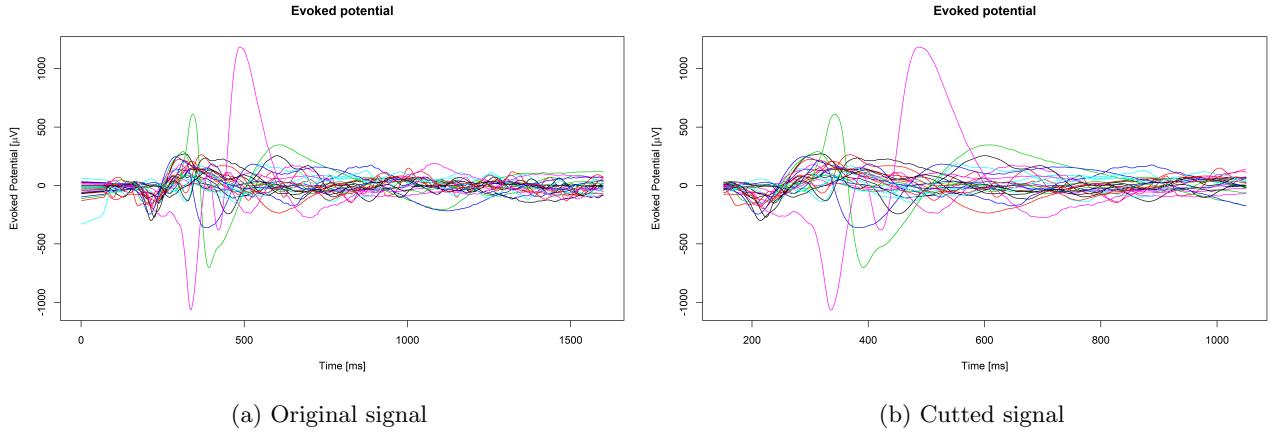


Figure 5: Left lobe short latency signal

5.1 Data Preprocessing and Basis Expansion

First, we proceed by preprocessing data at our disposal. In figure 5a, we observe that functions present long tails. We decide to focus on the central and most shape-characteristic part of our functions, keeping the points from the 150th to the 1500th (see figure 5b). Furthermore, due to computational reasons, we decide to keep one time point out of nine, getting finally 100 time points. On the other hand, in the context of dimensionality reduction (see Section 2.1), we opted for a regression splines approach for smoothing. We set the spline order of the spline basis to 4 and the number of the basis is chosen through a generalized cross validation criterion (GCV) (see Ramsay and Silverman 2005). We compute the GCV index for each function over a grid of possible number of basis (from 6 to 80) and then compute the mean for each possible number of basis over the different functions. The minimum value of the mean of the GCV index is attained for 80 basis, but we can appreciate from fig 6 that around 25 basis there is an elbow. We hence opt for a fourth-order spline basis of 25 basis functions. Finally, we re-scale functional data, dividing by the maximum value attained by all the observations.

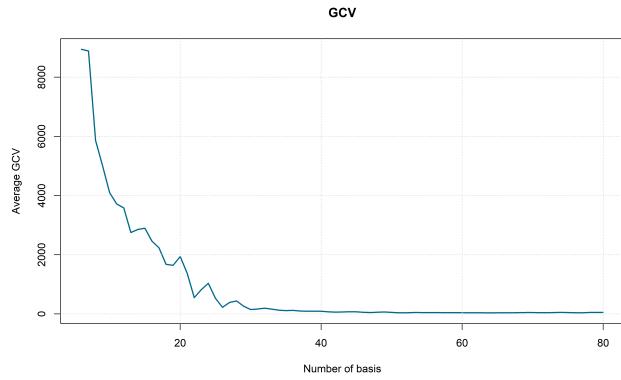
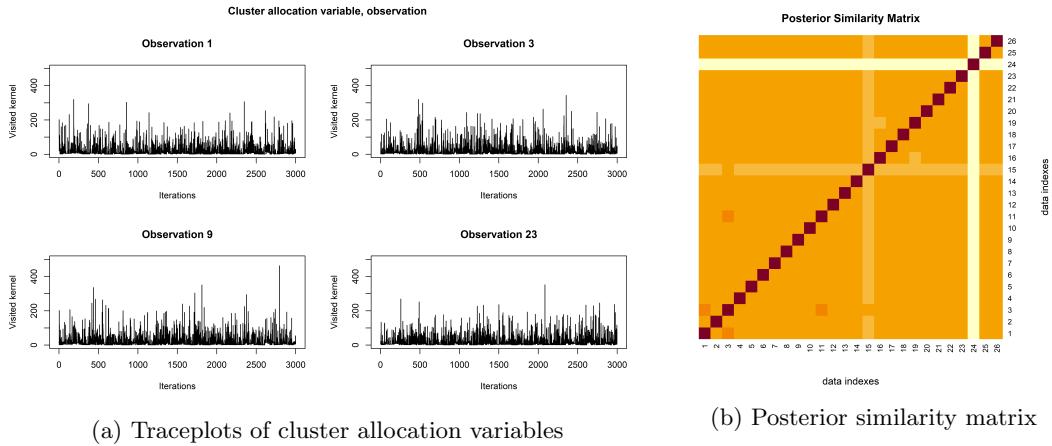


Figure 6: Mean of GCV index over the different functions for different possible number of basis

5.2 Testing

We performed 5,000 iterations after 15,000 burnin iterations. The truncation level of the stick-breaking sampler is fixed to 500 in order to encourage mixing. In [Figure 7a](#) we report traceplots of cluster allocation variable, in [Figure 7b](#) we present the posterior similarity matrix, while in [Figure 8](#) is reported the functional observations, coloring them by means of the optimal partition found.

We can appreciate that mixing performances are very satysfying, as highlighted by cluster allocation variables traceplots. Nonetheless, the posterior similarity matrix clearly shows how the algorithm is reluctant in separating observations in different clusters. We notice that only 1 out of 26 observations are separated from the others in the partition that minimizes the Binder loss function. Being this curve the most different in terms of range suggests that the algorithm is not really able to distinguish between the others. By repeating the analysis excluding such curve, we still notice that the algorithm is unable to separate the remaining functions in different clusters.



[Figure 7: Left lobe short latency signal: traceplots and PSM](#)

We repeat the analysis on all four components of the Sensorial Evoked Potential, observing the algorithm perform similarly. That is, it is only able to isolate the curves that are clearly different in terms of amplitude and phase with respect to the others. As usual, we report here posterior inference for the right lobe short latency (dxSL) signal, representing in [Figure 9a](#) we traceplots of cluster allocation variables. In [Figure 9b](#) is reported the posterior similarity matrix, while in [Figure 10](#) we visualize the functional observations, coloring them by means of the optimal partition found.

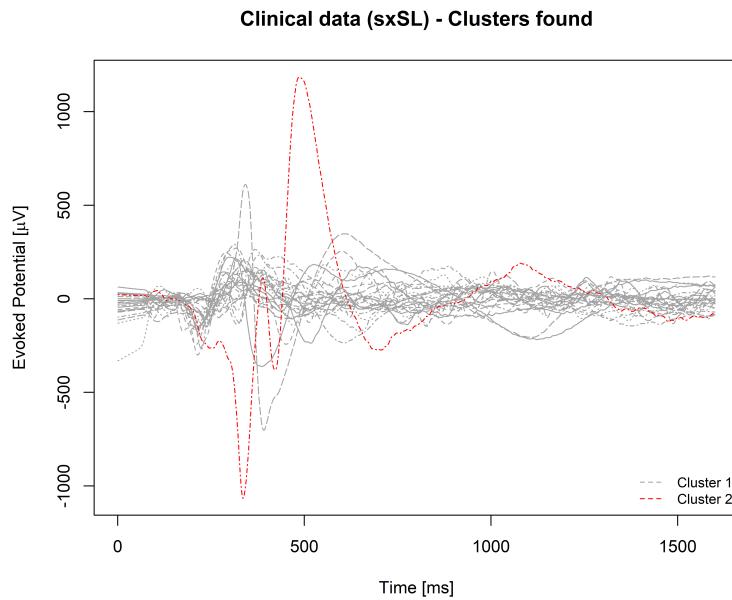


Figure 8: Left lobe short latency signal: clusters found

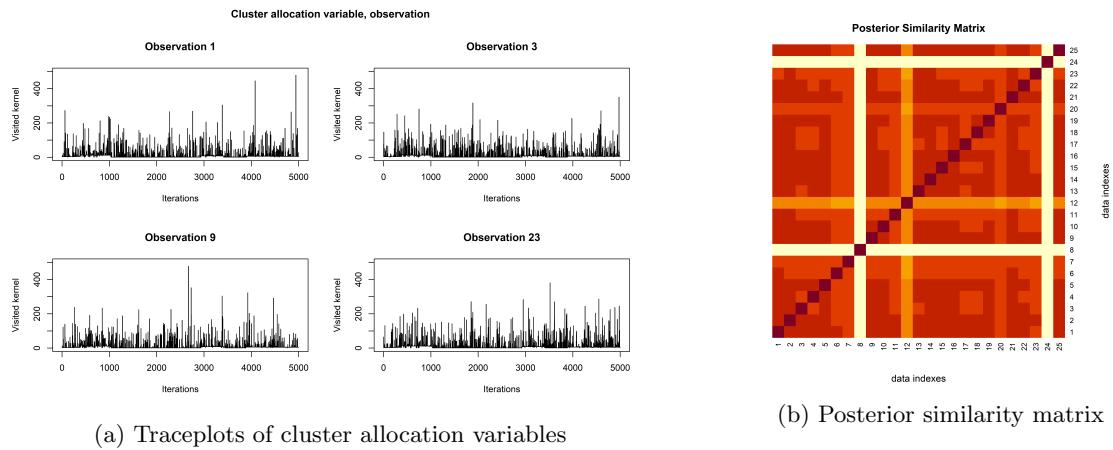


Figure 9: Right lobe short latency signal: traceplots and PSM

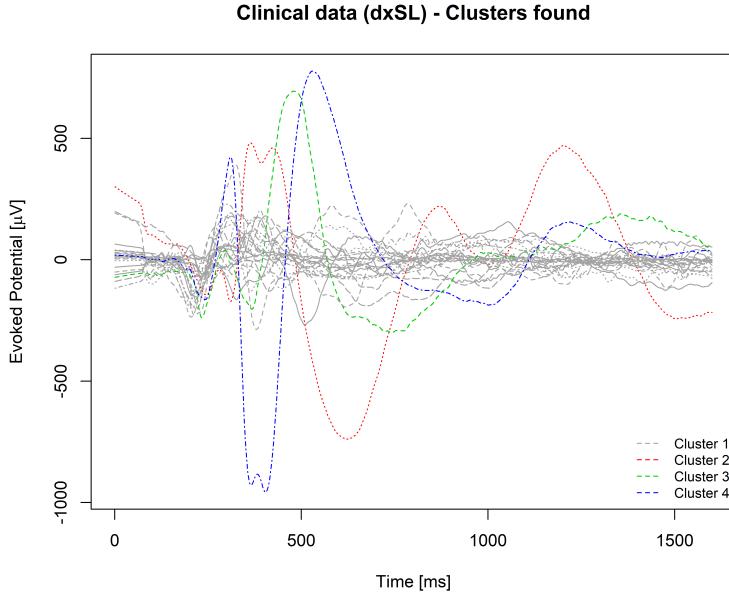


Figure 10: Right lobe short latency signal: clusters found

5.2.1 Proposal of a New Model

Given such poor performance in the clustering of observations, we extend the model by dropping the assumption of identical distribution for the $\phi_t \forall t$. Namely, we now assume $\forall t$:

$$\phi_t \stackrel{ind}{\sim} IG(c_t, d_t)$$

We implement a new sampling algorithm accordingly and test it both on simulated data and on our observations. Since the testing of the algorithm do not result in any improvement in the clustering performance, we decide to mention this extension only in this subsection, and not to give it any more emphasis.

5.3 Conclusions

The proposed algorithm has not been able to perform an effective clustering of clinical data. In particular, in all the four components no clustering structure is detected, except for the isolation of very dissimilar functions in shape and amplitude. We argue that this may be due to different reasons. First, our model may not be adequate for our real data. In particular, our data may be far from being effectively modeled by mixture of Gaussian Processes, especially with our assumptions on the form of the covariance operator. A possible research path may be to set a more complex form of the covariance operator, dropping for instance the assumption of independence between different time instants. On the other hand, a more complex hyperprior structure may be set, as done in Scarpa and Dunson 2014, where also a hyperprior is set on the parameters of the inv-gamma prior on ϕ_t (12). This has not been done for different reasons, among whom the problem of the absence of conjugacy that leads to the need of a Metropolis-Hastings step in the algorithm. The computation of the acceptance rate as in *ibid.* would have required the product of high number of factors that may lead to very low acceptance probabilities and so to increase substantially the computation time.

References

- David Kahle, James Stamey (2017). “invgamma: The Inverse Gamma Distribution”. In: URL: <https://CRAN.R-project.org/package=invgamma>.
- Ferguson, T.S. (1973). “A Bayesian analysis for some nonparametric problems”. In: *The annals of statistics* 1.2, pp. 209–230.
- Fritsch, Arno (2012). “mcclust: Process an MCMC Sample of Clusterings”. In: URL: <https://CRAN.R-project.org/package=mcclust>.
- Hadley Wickham Jim Hester, Winston Chang (2015). “devtools: Tools to Make Developing R Packages Easier”. In: URL: <https://CRAN.R-project.org/package=devtools>.
- Ieva, Francesca et al. (2019). “roahd Package: Robust Analysis of High Dimensional Data”. In: *The R Journal* 11.2, pp. 291–307. URL: <https://doi.org/10.32614/RJ-2019-032>.
- Ishwaran, Hemant and Lancelot F. James (2001). “Gibbs sampling methods for stick-breaking priors”. In: *Journal of the American Statistical Association* 96.453, pp. 161–173.
- J. O. Ramsay Spencer Graves, Giles Hooker (2020). “fda: Functional Data Analysis”. In: URL: <https://CRAN.R-project.org/package=fda>.
- Kevin Kuang Quyu Kong, Francesco Napolitano (2019). “pbmcapply Package: Tracking the Progress of Mc*pply with Progress Bar”. In: URL: <https://cran.r-project.org/web/packages/pbmcapply/index.html>.
- Meschiari, Stefano (2015). “latex2exp Package: Use LaTeX Expressions in Plots”. In: URL: <https://CRAN.R-project.org/package=latex2exp>.
- Mueller, P. et al. (2015). *Bayesian Nonparametric Data Analysis*. Springer.
- Neal, Radford M. (2000). “Markov chain sampling methods for Dirichlet process mixture models”. In: *Journal of Computational and Graphical Statistics* 9.2, pp. 249–265.
- Plummer, Martyn et al. (2006). “CODA: Convergence Diagnosis and Output Analysis for MCMC”. In: *R News* 6.1, pp. 7–11. URL: <https://journal.r-project.org/archive/>.
- R Core Team (2013). “R: A Language and Environment for Statistical Computing”. In: URL: <http://www.R-project.org/>.
- Ramsay, J. and B. Silverman (2005). “Functional Data Analysis”. In: Springer. Chap. 4.
- Scarpa, Bruno and David B. Dunson (2014). “Enriched stick-breaking processes for functional data”. In: *Journal of the American Statistical Association* 109.506, pp. 647–660.
- Statisticat and LLC. (2020). “LaplaceDemon: Complete Environment for Bayesian Inference”. In: R package version 16.1.4. URL: <https://web.archive.org/web/20150206004624/http://www.bayesian-inference.com/software>.
- Teh, Y. W (2010). “Dirichlet processes”. In: *Encyclopedia of Machine Learning*, Springer.
- Venables, W. N. and B. D. Ripley (2002). “Modern Applied Statistics with S”. In: ISBN 0-387-95457-0. URL: <https://www.stats.ox.ac.uk/pub/MASS4/>.
- Wade, Sara (2015). “mcclust.ext: Process an MCMC Sample of Clusterings”. In: URL: <https://github.com/sarawade/mcclust.ext>.
- Wickham, Hadley (2011). “The Split-Apply-Combine Strategy for Data Analysis”. In: *Journal of Statistical Software* 40.1, pp. 1–29. URL: <http://www.jstatsoft.org/v40/i01/>.

Code

All the analysis were implemented in

R Core Team (2020). R: A language and environment for statistical computing.
R Foundation for Statistical Computing, Vienna, Austria.
URL <https://www.R-project.org/>.

Codes are publicly available in this [Github repository](#).

The code is structured as follows:

- main.R: R-script containing all the steps for running the Blocked Gibbs sampler
- Tools: folder containing all the script needed in main.R, among whom, FBNP.R containing a function that implements Gibbs sampler without hyperpriors and FBNP_hyper.R containing a function that implements Gibbs Sampler with hyperpriors
- Posterior inference: folder containing the script main_posterior_inference.R for performing posterior inference for the latent partition and convergence diagnostic and all the script needed in the main one
- Gaussian process: folder containing the scripts main_GP_indep.R and main_GP_exp.R that perform the Blocked Gibbs sampler on the simulated data with diagonal and exponential covariance operator and all the needed script
- Alltime: folder containing main_alltime.R that performs the Blocked Gibbs sampler changing the prior structure on the covariance operator, setting a different prior on different time instants

The script install.R provides automatic installation of required R packages.

A Full Conditional for ϕ_t

Fixing $t \in \{t_0, \dots, T\}$, we have

$$f(\phi_t | \text{rest}) \propto H(\phi_t) \prod_{\{i:K_i=j\}} f(X_i(t) | Z_j, Y_j).$$

Recall that prior for ϕ_t is $\forall t \phi_t \stackrel{iid}{\sim} IG(c, d)$, thus we can derive:

$$\begin{aligned} f(\phi_t | \text{rest}) &\propto (\phi_t)^{-c-1} e^{-d/\phi_t} \prod_{\{i:K_i=j\}} (\phi_t)^{-1/2} \exp \left\{ -\frac{(X_i(t) - \mu(t))^2}{2\phi_t} \right\} \\ &\propto (\phi_t)^{-c-1} e^{-d/\phi_t} (\phi_t)^{-r/2} \exp \left\{ -\frac{1}{\phi_t} \sum_{\{i:K_i=j\}} \frac{(X_i(t) - \mu(t))^2}{2} \right\} \\ &\propto (\phi_t)^{-(c+r/2)-1} \exp \left\{ -\frac{1}{\phi_t} \left[d + \sum_{\{i:K_i=j\}} \frac{(X_i(t) - \mu(t))^2}{2} \right] \right\} \end{aligned}$$

Finally, one can identify full conditional for ϕ_t as:

$$\phi_t | \text{rest} \sim \text{inv-gamma} \left(c + \frac{r}{2}, d + \sum_{\{i:K_i=j\}} \frac{(X_i(t) - \mu(t))^2}{2} \right)$$

B Full Conditional for μ

We have

$$f(\boldsymbol{\mu} | \text{rest}) \propto H(\boldsymbol{\mu} | Y_j) \prod_{\{i:K_i=j\}} f(\mathbf{X}_i | Z_j, Y_j)$$

For what concerns the first term, we recall that the prior for $\boldsymbol{\mu}$ is $\boldsymbol{\mu} | \mathbf{m}_0, \Lambda_0 \sim \mathcal{N}_L(\mathbf{m}_0, \Lambda_0)$.

Aiming to express the likelihood of X_i in terms of β_i , the vector of coefficients of the projection in basis, we exploit the basis expansion introduced in 2.1, in particular we recall that:

$$X_i(t) = \mathbf{b}(t)^T \cdot \boldsymbol{\beta}_i \quad \mu(t) = \mathbf{b}(t)^T \cdot \boldsymbol{\mu}$$

This allows to rewrite the density of $\mathbf{X}_i | Z_j, Y_j$ in the following way:

$$\begin{aligned} f(\mathbf{X}_i | Z_j, Y_j) &\propto \prod_{t=t_0}^T \exp \left\{ -\frac{(X_i(t) - \mu(t))^2}{2\phi_t} \right\} \\ &\propto \exp \left\{ -\frac{1}{2} \sum_{t=t_0}^T \frac{(X_i(t) - \mu(t))^2}{\phi_t} \right\} \\ &\propto \exp \left\{ -\frac{1}{2} \sum_{t=t_0}^T \frac{(\mathbf{b}(t)^T \boldsymbol{\beta}_i - \mathbf{b}(t)^T \boldsymbol{\mu})^2}{\phi_t} \right\} \\ &\propto \exp \left\{ -\frac{1}{2} (\boldsymbol{\beta}_i - \boldsymbol{\mu})^T \sum_{t=t_0}^T \frac{\mathbf{b}(t) \mathbf{b}(t)^T}{\phi_t} (\boldsymbol{\beta}_i - \boldsymbol{\mu}) \right\} \end{aligned}$$

The full conditional for μ has the following form:

$$\begin{aligned} f(\boldsymbol{\mu}|\text{rest}) &\propto \mathcal{N}_L(\mathbf{m}_0, \Lambda_0) \prod_{\{i:K_i=j\}} f(\mathbf{X}_i|Z_j) \\ &\propto \exp \left\{ -\frac{1}{2}(\boldsymbol{\mu} - \mathbf{m}_0)^T \Lambda_0^{-1} (\boldsymbol{\mu} - \mathbf{m}_0) \right\} \prod_{\{i:K_i=j\}} \exp \left\{ -\frac{1}{2}(\boldsymbol{\beta}_i - \boldsymbol{\mu})^T \sum_{t=t_0}^T \frac{\mathbf{b}(t)\mathbf{b}(t)^T}{\phi_t} (\boldsymbol{\beta}_i - \boldsymbol{\mu}) \right\} \end{aligned}$$

Dropping $-\frac{1}{2}$ in the exponential argument and setting $B := \sum_{t=t_0}^T \frac{\mathbf{b}(t)\mathbf{b}(t)^T}{\phi_t}$ we obtain:

$$\begin{aligned} &(\boldsymbol{\mu} - \mathbf{m}_0)^T \Lambda_0^{-1} (\boldsymbol{\mu} - \mathbf{m}_0) + \sum_{\{i:K_i=j\}} (\boldsymbol{\beta}_i - \boldsymbol{\mu})^T B (\boldsymbol{\beta}_i - \boldsymbol{\mu}) \\ &\propto \boldsymbol{\mu}^T \Lambda_0^{-1} \boldsymbol{\mu} - 2\boldsymbol{\mu}^T \Lambda_0^{-1} \mathbf{m}_0 + \boldsymbol{\mu}^T \sum_{\{i:K_i=j\}} B \boldsymbol{\mu} - 2\boldsymbol{\mu}^T B \sum_{\{i:K_i=j\}} \boldsymbol{\beta}_i \\ &\propto \boldsymbol{\mu}^T [\Lambda_0^{-1} + rB] \boldsymbol{\mu} - 2\boldsymbol{\mu}^T \left(\Lambda_0^{-1} \mathbf{m}_0 + B \sum_{\{i:K_i=j\}} \boldsymbol{\beta}_i \right) \\ &\propto \boldsymbol{\mu}^T \Lambda_r^{-1} \boldsymbol{\mu} - 2\boldsymbol{\mu}^T \Lambda_r^{-1} \Lambda_r \left(\Lambda_0^{-1} \mathbf{m}_0 + B \sum_{\{i:K_i=j\}} \boldsymbol{\beta}_i \right) \\ &\propto (\boldsymbol{\mu} - \mathbf{m}_r)^T \Lambda_r^{-1} (\boldsymbol{\mu} - \mathbf{m}_r) \end{aligned}$$

where

$$\begin{aligned} \mathbf{m}_r &= \Lambda_r \left(\Lambda_0^{-1} \mathbf{m}_0 + \left[\sum_{t=t_0}^T \frac{\mathbf{b}(t)\mathbf{b}(t)^T}{\phi_t} \right] \sum_{\{i:K_i=j\}} \boldsymbol{\beta}_i \right) \\ \Lambda_r &= \left(\Lambda_0^{-1} + r \left[\sum_{t=t_0}^T \frac{\mathbf{b}(t)\mathbf{b}(t)^T}{\phi_t} \right] \right)^{-1} \end{aligned}$$

Given these calculations, we conclude that the full conditional for $\boldsymbol{\mu}$ is:

$$\boldsymbol{\mu}|\text{rest} \sim \mathcal{N}_L(\mathbf{m}_r, \Lambda_r) \tag{26}$$