POLITECNICO DI MILANO

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

A Bayesian nonparametric model for density and cluster estimation: the ε-NGG mixture model

Relatrice: Prof.sa Alessandra Guglielmi
Correlatore: Dr. Raffaele Argiento

Tesi di Laurea di:
Ilaria Bianchini
Matr. 783194

Anno Accademico 2012/2013

Sommario

Il presente lavoro di tesi tratta della definizione e dell'applicazione di una nuova classe di prior non parametriche che approssima un processo appartenente alla famiglia di misure aleatorie normalizzate a incrementi indipendenti. Queste ultime sono misure di probabilità aleatorie discrete i cui pesi, che sono infiniti, sono ottenuti mediante normalizzazione dei salti di un processo di Poisson, mentre i punti di supporto sono un insieme numerabile di variabili aleatorie indipendenti e identicamente distribuite da un certa legge. La particolare classe di prior non parametriche che vogliamo approssimare in questa tesi è il processo gamma generalizzato normalizzato (NGG).

L'inferenza è complicata a causa della presenza di infiniti parametri non noti, che sono i pesi e il supporto della misura aleatoria discreta. Per risolvere ciò saranno tenuti in considerazione nel processo solo i salti del processo NGG maggiori di una certa soglia ε: tale definizione rende la prior di dimensione finita. Il parametro ε controlla il livello di approssimazione, da cui il nome di processo ε-NGG. Successivamente, il nuovo processo verrà considerato come misura misturante di un modello mistura, spesso usato in statistica bayesiana non parametrica come un modello flessibile per problemi di stima di densità e *clustering*.

In questa tesi costruiremo un algoritmo Gibbs sampler per simulare dalla posterior del modello mistura. L'algoritmo verrà poi applicato a due diversi dataset. Il primo dataset, univariato, è il ben noto dataset *Galaxy*, mentre il secondo, multivariato, è chiamato in letteratura *Yeast cell cycle* dataset e raccoglie i profili di espressione genetica in 9 diversi istanti di tempo. Per entrambi condurremo un'approfondita analisi di robustezza rispetto alla scelta della prior per valutare sia la bontá del modello in un contesto di stima di densitá, sia l'influenza dei parametri, i quali possono anche essere considerati aleatori, sulle stime. Nel caso multivariato, infine, il nostro processo verrá inserito all'interno di un modello di *clustering*: per scegliere la migliore stima a posteriori sará usato il metodo di minimizzazione della funzione di perdita.

Abstract

In this work we define a new class of random probability measures, approximating the well-known normalized generalized gamma (NGG) process. Our new process is defined from the representation of NGG processes as discrete measures where the weights are obtained by normalization of the jumps of a Poisson process, and the support consists of independent and identically distributed (iid) points, however considering only jumps larger than a threshold ε. Therefore, the number of jumps of this new process, called ε-NGG process, is a.s. finite. A prior distribution for ε can be elicited. We will assume the ε-NGG process as the mixing measure in a mixture model for density and cluster estimation. Moreover, an efficient Gibbs sampler scheme to simulate from the posterior is provided. The model is then applied to two datasets, the well-known univariate Galaxy dataset and the multivariate Yeast cell cycle dataset, consisting of gene expression profiles measured at 9 different times. A deep robustness analysis with respect to the prior is performed for both models, in order to evaluate the goodness-of-fit of the model in a density estimation context and investigate the role of the parameters (which can also be considered as random variables) in the posterior estimates. In the multivariate case, we will also provide posterior cluster estimates, obtained through a loss-function minimization approach.

Contents

List of Figures

List of Tables

Introduction

Sometimes in either density estimation and clustering problems a parametric approach could be too restrictive, leading to wrong inference and decisions: an approach allowing for a richer and larger class of models is needed. This is achieved by considering infinite dimensional families of probability models. Priors on such families are known as nonparametric Bayesian priors. In this work we propose a mixture model that considers as mixing measure an homogeneous normalized random measure with independent increments, in particular the normalized generalized gamma (NGG) process. This random probability measure is more flexible than the Dirichlet process, considering a wider range of conditions. In general, the main difficulty of the nonparametric approach is that posterior inference involves the computation of infinite unknown parameters; in the literature, there are two main approaches to deal with this problem, namely marginal and truncation algorithms. The former integrate out the infinite dimensional parameter (i.e. the random probability), while the latter ones approximate the infinite dimensional process with a finite dimensional one, yielding a full Bayesian analysis. This represents a positive aspect of the algorithm, since also estimates of all the parameters of the random process can be obtained.

The solution we propose here can be classified as an a-priori truncation method, similar to the approach of Ishwaran and James (2001) for the DPM model. The main original contribution of the thesis, in fact, is the definition of a new discrete random probability measure, called ε-NGG process, which is a truncated version of the NGG process; in this sense, a convergence result is also provided. Another achievement of the work is the construction of a Gibbs sampler scheme to simulate from the posterior of the ε-NGG mixture model; in particular, we have built a conditional algorithm that uses a finite random number of parameters, but, on the other hand, it is easy to implement. We coded it in C++ language, while the post-processing has been developed with the software R. We have applied our model to two popular datasets: the Galaxy data, since it is nowadays the favorite univariate test dataset for any new nonparametric model in a density estimation context, and the Yeast

cell cycle data, which is an interesting multivariate dataset consisting of gene expression profiles measured at 9 different times.

Density estimates for the two applications are shown, together with a deep robustness analysis of the estimates and the convergence of posterior chains for different prior choices; moreover, a prior on the parameters of the process NGG and on ε can be elicited in order to robustify the inference. We deduce from the analysis that the model is robust with respect to the choice of the scalar parameters of the NGG process, while it depends strongly on the choice of the distribution P_0, that controls the support of the prior distribution. This issue is important, especially in the multivariate case, where the parameters have to be carefully selected. On the one hand, in the univariate application, it is possible to choose a relatively large ε, thus reducing the computational efforts but still obtaining reliable estimates. On the other hand, ε strongly affects the results in the multivariate case. Hence, smaller values for the parameter are needed in order to have a close approximation of the elicited nonparametric prior. We point out that the algorithm is quite fast, especially in the univariate case, where no multivariate sampling is needed.

Furthermore, when considering a sample from a discrete random probability measure, as in this case, ties among the sampled values yield a (random) partition π of the sample labels. Therefore, we have used the hierarchical structure of our model in order to provide the clustering of data; in particular, we have chosen a posterior estimate of π through a loss function minimization method. Finally, we have also illustrated the performances of the clustering algorithm in the multivariate case.

The work is organized as follows: in Chapter 1 the class of NRMI priors and mixture models for density estimation are introduced, with focus on the NGG process. Chapter 2 deals with the definition of the ε-NGG process. First, we prove the convergence in law to the NGG process when ε tends to 0; then, the Gibbs sampler algorithm for the posterior inference from an ε-NGG mixture model is provided. The algorithm is applied to the univariate Galaxy dataset in Chapter 3: a robustness analysis is carried on, illustrating the results of our estimates for different sets of hyperparameters, in order to understand the effect of the parameters on posterior estimates. In Chapter 4 a similar analysis is provided on a multivariate dataset: the Yeast cell cycle data. In this context, we also tackle the clustering problem.

Chapter 1 | Density estimation by NRMI mixtures

In this first chapter we deal with the problem of density estimation from a Bayesian nonparametric point of view. The nonparametric approach is very useful because it allows a rich class of models for the data: this is achieved by considering infinite dimensional families of probability models. Priors on such families are known as nonparametric Bayesian priors and prevent from misleading decisions and inference that may be done in the parametric approach, which considers families of models that can be indexed by a finite dimensional set of parameters. The parametric paradigm requires a strong knowledge of the phenomenon taken into account and, as clearly explained in Müller and Mitra (2013), can mislead investigators into a false sense of posterior certainty.

In this chapter we will see how a particular family of nonparametric priors, namely normalized random measures with independent increments (NRMI), can be included as an ingredient in density estimation problems. The main tool is the mixture model of Section 1.1.

1.1 | Mixture models

Let us start recalling the notion of exchangeability. Let $(X_n)_{n\geq 1}$ be a sequence of observations, defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where each X_i takes values in $\mathbb{X}$, a complete and separable metric space endowed by a σ-algebra $\mathcal{X}$ (for instance $\mathbb{X} = \mathbb{R}^k$ for some positive integer k and $\mathcal{X} = \mathcal{B}(\mathbb{R}^k)$).

The typical assumption in the Bayesian approach is exchangeability of infinite sequences of data. Formally, this means that for every $n \geq 1$ and every permutation $\pi(\cdot)$ of the indices 1, 2,.., n, the probability distribution of the vector $(X_1, X_2, \ldots, X_n)$ is equal to that of $(X_{\pi(1)}, X_{\pi(2)}, .., X_{\pi(n)})$.

Before stating the theorem allowing the formalization of the nonparametric model, the *de Finetti's theorem*, let us define a random probability

measure as

$$P : (\Omega, \mathcal{F}) \to (\mathcal{P}_{\mathbb{X}}, \mathcal{C}_{\mathcal{P}})$$

where $\mathcal{P}_{\mathbb{X}}$ is the space of all the probability measures on $(\mathbb{X}, \mathcal{X})$ and $\mathcal{C}_{\mathcal{P}}$ is the smallest σ-algebra such that $P \mapsto P(B)$ is a measurable function, for any $B \in \mathcal{X}$.

The *de Finetti's representation theorem* states that the sequence $(X_n)_{n\geq 1}$ is exchangeable if and only if there exists a random probability measure Q on the space of the probability measures on $\mathbb{X}$ such that

$$\begin{cases} X_i|P \overset{iid}{\sim} P & i = 1, 2, .., n \\ P \sim Q \end{cases}$$

for any $n \geq 1$. The random element P is defined on $(\Omega, \mathcal{F}, \mathbb{P})$ with values in $\mathcal{P}_{\mathbb{X}}$ and the distribution Q is the so-called *de Finetti measure* and represents the prior distribution. If Q concentrates all the mass over a family of distributions, namely the population distribution, that can be indexed by a parameter of finite dimension, then the inferential problem is called *parametric*, otherwise the problem is *nonparametric*.

Mixture models provide a statistical framework for modeling a collection of continuous observations $(X_1, \ldots, X_n)$ where each measurement is supposed to arise from one of k groups, with k eventually unknown, and each group is modeled by a kernel distribution from a suitable parametric family.

This model is usually represented hierarchically in terms of a collection of independent and identically distributed latent random variables $(\theta_1, \ldots, \theta_n)$:

$$\begin{cases} X_i|\theta_i \overset{ind}{\sim} K(\cdot|\theta_i) & i = 1, \ldots, n \\ \theta_i|P \overset{iid}{\sim} P & i = 1, \ldots, n \\ P \sim Q \end{cases} \tag{1.1}$$

where Q denotes the nonparametric prior distribution and $K(\cdot|\theta)$ is a probability density function parametrized by the latent random variable θ.

Model (1.1) is equivalent to assume the data $X_1, \ldots, X_n$ as i.i.d. according to a probability density that is a mixture of kernel functions:

$$X_1, \ldots, X_n \overset{iid}{\sim} f(x) = \int_{\Theta} K(x|\theta) P(d\theta),$$

where P is called mixing measure. Note that if Q selects discrete probability measures, P is discrete and the mixture model can be written as a sum with a countably infinite number of components:

$$f(x) = \sum_{j=1}^{+\infty} p_j K(x|\theta_j)$$

where the weights $(p_j)_{j\geqslant 1}$ represent the relative frequency of the groups in the population indexed by θ_j.

This approach provides a flexible model for clustering items in a hierarchical setting without the necessity to specify in advance the exact number of clusters. This fact will be explained clearly later on.

The most popular model of this family is the Dirichlet Process Mixture (DPM) model where the random probability measure Q is indeed the Dirichlet process. In what follows, we introduce a more general class of mixture models, namely mixtures with mixing measure given by normalized random measures with independent increments (NRMI), since they include the DPM as a specific case. As we will see, the NRMIs are very flexible but still mathematically tractable, making them a good choice as Q in the mixture models. In the next section completely random measures are introduced: they are the basic block to construct our nonparametric priors.

1.2 | Completely random measures

We refer to Kingman (1993) for all the material in this section.

First let us introduce the Poisson process and the associated count function: they will be useful later in this section.

Definition 1 (Poisson Process)**.** *A Poisson process* Π *on* $\mathbb{X}$ *is a random countable subset of* $\mathbb{X}$ *such that:*

1. *for any disjoint numerable subsets* $A_1, A_2, \ldots, A_n$ *of* $\mathbb{X}$*, the random variables* $N(A_1), N(A_2), \ldots, N(A_n)$ *are independent,*

2. $N(A)$ *has the Poisson distribution* $\mathcal{P}(\nu)$*, where* $\nu = \nu(A)$ *is such that* $0 \leqslant \nu \leqslant \infty$*.*

We denote by N the cardinality of the set $\Pi \cap A$, N(A)=$\sharp\{\Pi \cap A\}$, where A is a subset of the space $\mathbb{X}$ where the process takes place.

The measure ν totally characterizes the process: it is called *mean measure* or *Lévy's intensity*.

Let $(\mathbb{M}, \mathcal{B}(\mathbb{M}))$ be the space of boundedly finite measures on $(\mathbb{X}, \mathcal{B}(\mathbb{X}))$, i.e. for any measure μ in $\mathbb{M}$ and any bounded set A in $\mathbb{X}$, $\mu(A) < \infty$.

Definition 2 (Completely Random Measure, CRM). *A random element μ defined on $(\Omega, \mathcal{F}, \mathbb{P})$ and taking values in $\mathbb{M}$ is a completely random measure if, for any $A_1, A_2, \ldots, A_n$ in $\mathbb{X}$ with $A_i \cap A_j = \varnothing$ for any $i \neq j$ and $n \geqslant 1$, the random variables $\mu(A_1), \mu(A_2), \ldots, \mu(A_n)$ are mutually independent and*

$$\mu\left(\bigcup_{j\geq 1} A_j\right) = \sum_{j\geq 1} \mu\left(A_j\right) \qquad a.s.$$

In general, a CRM can be decomposed into three independent components: a non-random measure, a countable collection of non-negative random masses at non-random locations and a countable collection of non-negative random masses at random locations (see Chapter 8 of Kingman (1993) for a more detailed explanation). For our purposes we consider CRMs consisting only of the third component: they are discrete measures a.s., so μ can be written as an infinite weighted sum of Dirac functions:

$$\mu(\cdot) = \sum_{i\geq 1} J_i \delta_{\tau_i}(\cdot). \tag{1.2}$$

The random points $(J_i, \tau_i)_{i\geqslant 1}$ are the points of a Poisson process on $(\mathbb{R}^+, \mathbb{X})$ with mean measure ν that must satisfy the following conditions :

$$\int_{(0,1)} s\nu(ds, \mathbb{X}) < +\infty, \qquad \nu([1,\infty) \times \mathbb{X}) < +\infty. \tag{1.3}$$

This construction produces the most general completely random measure without fixed atoms and non-random measure.

An important property one could require from a CRM is the homogeneity: this property relies on the possibility to factorize the underlying mean measure. Let P_0 be a non-atomic and σ-finite probability measure on $\mathbb{X}$, then we have:

- if $\nu(ds, dx) = \rho(ds)P_0(dx)$, for some measure ρ on $\mathbb{R}^+$, we call N and μ *homogeneous*: in this case the jumps in the representation (1.2) are independent of the locations;
- if $\nu(ds, dx) = \rho(ds|x)P_0(dx)$, where $\rho : \mathcal{B}(\mathbb{R}^+) \times \mathbb{X} \to \mathbb{R}^+$, i.e. $x \mapsto \rho(C|x)$ is $\mathcal{B}(\mathbb{X})$ measurable for any C $\in \mathcal{B}(\mathbb{R}^+)$ and $\rho(\cdot|x)$ is a σ-finite measure on $\mathcal{B}(\mathbb{R}^+)$ for any x $\in \mathbb{X}$, we call N and μ *non homogeneous*.

The sequence $(J_i)_{i\geq 1}$ represents the jumps controlled by the kernel ρ and $(\tau_i)_{i\geq 1}$ are the locations of the jumps determined by the measure P_0 on $\mathbb{X}$.

See Figure 1.1 for a graphical example of an homogeneous CRM on $\mathbb{R}$ (left) with its corresponding Lèvy's intensity (right).

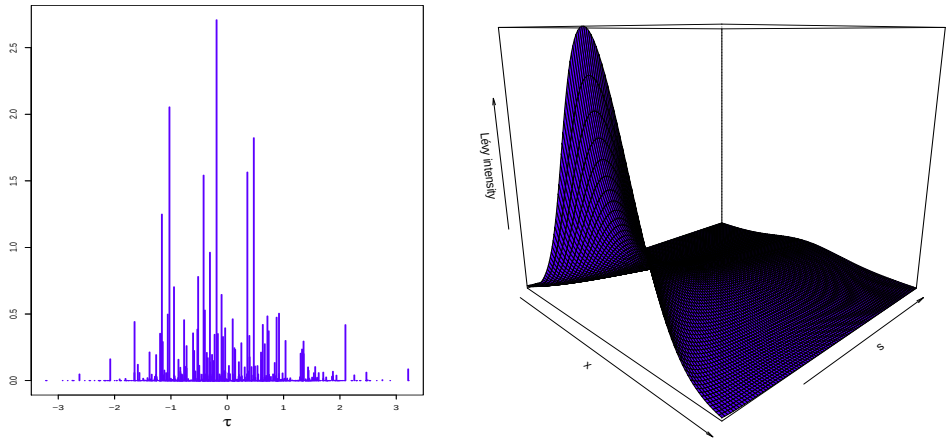


Figure 1.1: Left: A draw $\sum_{i>0} J_i \delta_{\tau_i}$ from a homogeneous CRM on $\mathbb{R}$. Each stick denotes an atom in the CRM with mass given by its height J_i and location given by τ_i. Right: the density of its Lèvy intensity measure $\nu(ds, dx) = 1/\Gamma(1-\sigma)e^{-s}s^{-1-\sigma}dsP_0(dx)$ where $\sigma = 0.1$ and P_0 is gaussian of mean 0 and variance 1.

Since μ is a discrete random measure almost surely, it is straightforward to build a discrete random probability measure by the normalization procedure, which yields to NRMIs, first introduced by Regazzini et al. (2003).

1.3 | Species sampling models

We introduce in this section the *species sampling* model which will be useful later, dealing with the process proposed in this work, the ε-NGG process. We refer to Pitman (1996) for all the results in this section.

A sequence of random variables $(\theta_n)_{n\geqslant 1}$ is a *species sampling sequence* if and only if $(\theta_n)_{n\geqslant 1}$ is a sample from a random distribution Q of the form

$$Q = \sum_i P_i \delta_{\tau_i} + \left(1 - \sum_i P_i\right) \eta$$

for some sequence of variables $(P_i)_{i\geqslant 1}$ such that

$$P_i > 0 \quad \forall i \quad \textit{and} \quad \sum_i P_i \leqslant 1 \quad a.s.$$

and some sequence $(\tau_i)_{i\geqslant 1}$ that is iid from η and independent of $(P_i)_{i\geqslant 1}$.

If θ_n represents the "species" (or the label) of the $n-th$ individual in some process of sampling of elements from a population, P_i can be interpreted as the relative frequency of the $i-th$ species and τ_i as the label assigned to that species. The model is *proper* if $\sum_i P_i = 1$ a.s., i.e. Q is almost surely discrete.

This class of random probability measures is characterized by a distribution representing the prior guess of the shape of the random measure, η, and a symmetric function of sequence of positive integers called exchangeable partition probability function (eppf). Before defining the eppf, it is useful to introduce the following notation: the finite sample $(\theta_1, \ldots, \theta_n)$ from a species sampling model Q induces a random partition $\mathbf{p}_n := \{C_1, \ldots, C_k\}$ on the set $\mathbb{N}_n := \{1, \ldots, n\}$ by letting $C_j = \{i : \theta_i = \theta_j^*\}$ for $j = 1, \ldots, k$, where θ_j^* are the unique values. In particular, $\#C_i = n_i$ for $1 \leq i \leq k$, and the eppf p can be viewed as a probability law on the set of all the partitions of $\mathbb{N}_n$. Recalling the definition given in Pitman (2003), an *exchangeable partition probability function* is a symmetric function p of sequences of positive integers $(n_1, \ldots, n_k)$ such that:

$$\mathbb{P}\left(\mathbf{p}_n = \{C_1, \ldots, C_k\}\right) = p\left(n_1, \ldots, n_k\right).$$

The marginal law of $(\theta_1, \ldots, \theta_n)$ has a unique characterization in term of its unique values $\boldsymbol{\theta}^* := (\theta_1^*, \ldots, \theta_k^*)$ and its corresponding exchangeable partition $\mathbf{p}_n$ given by

$$\mathcal{L}(\mathbf{p}_n, \theta_1^*, \ldots, \theta_k^*) = p(n_1, \ldots, n_k) \prod_{j=1}^{k} \mathcal{L}(\theta_j^*),$$

where p is the eppf associated to Q.

For a proper sequence $(P_i)_{i \geqslant 1}$, the following formula for the corresponding eppf is valid:

$$p(n_1, \ldots, n_k) = \sum_{j_1, \ldots, j_k} \mathbb{E}\left(\prod_{i=1}^{k} P_{j_i}^{n_i}\right), \tag{1.4}$$

where $(j_1, \ldots, j_k)$ ranges over all permutations of k positive integers.

In Section 2.2 we will see an analytical result on the eppf of the process P_ε we are going to introduce, which is exactly a species sampling model.

1.4 | Normalized random measures with independent increments

After the definition of species sampling models, we can resume the construction of NRMIs from the notion of completely random measures. First notice that the normalization procedure is well defined only if the total mass of the measure $T := \mu(\mathbb{X})$ is positive and finite almost surely:

$$\mathbb{P}\left(0 < T < +\infty\right) = 1 \qquad a.s.$$

This requirement is satisfied if the intensity ν in the more general non homogeneous case is such that

$$\int_{\mathbb{R}^+} \rho(ds|x) = +\infty \quad \forall x \in \mathbb{X}. \tag{1.5}$$

This means that the jumps of the process form a dense set in $(0, +\infty)$. Note that there are infinite masses near the origin since the second condition in (1.3) must hold. Now we can proceed with the definition of a NRMI.

Definition 3 (Normalized random measure with independent increments, NRMI)**.** *Let μ be a CRM with intensity measure ν such that $0 < \mu(\mathbb{X}) < \infty$ almost surely. Then, the random probability measure*

$$P(\cdot) = \frac{\mu(\cdot)}{\mu(\mathbb{X})}$$

is called normalized random measure with independent increments, NRMI, on $(\mathbb{X}, \mathcal{B}(\mathbb{X}))$.

It is important to highlight that NRMIs select, almost surely, discrete distributions, so that P admits a series representation as

$$\sum_{j \geq 1} p_j \delta_{\tau_j}$$

where $p_j = J_j/T \ \forall j \geqslant 1$ where the weights J_j are those in (1.2).

In order to understand better how the jumps and the locations derive from the homogeneous mean measure ν of the underlying CRM, we recall the definition of Bernoulli process.

Let Π be a Poisson process on a space $\mathbb{Y}$ with mean measure ν such that $\nu(\mathbb{Y}) < +\infty$: then Π is a finite subset of $\mathbb{Y}$ almost surely, since the total

number of points $N(\mathbb{Y})$ has the Poisson distribution $\mathcal{P}(\nu(\mathbb{Y}))$. What happens if we condition the process Π on the value $N(\mathbb{Y})$, the total number of points? Kingman (1993) explains that, given $N(\mathbb{Y})$, the points of a finite Poisson process are $N(\mathbb{Y})$ independent random variables with common distribution

$$q(\cdot) = \frac{\nu(\cdot)}{\nu(\mathbb{Y})}.$$

This important result allows us to state that, conditioning on a (finite) number of points, the random locations τ_js can be considered to be i.i.d.according to the probability measure P_0 of the factorization $\nu(ds, dx) = \rho(ds)P_0(dx)$ and it would be the same in the case of non-homogeneous measures. Note that if the support of P_0 coincides with $\mathbb{X}$, then the corresponding NRMI has full support $\mathcal{P}_{\mathbb{X}}$, which is a desirable property for a prior distribution.

On the other hand, the distribution of the random jumps $(J_j)_{j\geq 1}$ is governed by a Poisson process with Lèvy measure ρ which must satisfy condition (1.5) in order to be able to normalize the jumps. These jumps are infinite, therefore it is not possible to consider a Bernoulli process. In the next chapter we will see a way to sample these J_js. As we have already mentioned, if the underlying intensity ν is homogeneous, the factorization implies that the weights $p_i s$ are independent from the locations τ_i. Note also that P is a *proper species sampling model*.

A challenging issue when dealing with NRMIs in a statistical framework is the computation of posterior distribution, because the NRMIs are not conjugate, with the exception of the Dirichlet process. As shown in James et al. (2009), conditioning on a specific latent variable, the posterior distribution of a NRMI coincides with the distribution of another NRMI having a rescaled intensity and fixed points of discontinuity.

This can be considered as a kind of *conditional conjugacy* property, that makes the computation simpler. In particular, we define a positive random variable U as follows: let Γ_n be a Gamma random variable with shape and scale parameters n and 1, respectively, independent from the total sum T. Setting $U = \Gamma_n/T$ and conditioning with respect to U, the conjugacy is retrieved. It is immediate to show that for $n \geqslant 1$ the density function of U is given by

$$f_U(u) = \frac{u^{n-1}}{\Gamma(n)} \int_{\mathbb{R}^+} t^n e^{-ut} f_T(t) dt$$

where f_T is the density function of T.

The main result concerning a posterior characterization of the completely random measure is the following:

Theorem 1. *Let $(X_1, \ldots, X_n)$ be a sample from P, where P is a NRMI with intensity $\nu(ds, dx) = \rho(ds|x)P_0(dx)$. Then the CRM conditioned to the variable U and the sample is the sum of two measures*

$$\mu^{|(U,X_1,..,X_n)} \stackrel{d}{=} \mu^{|U} + \sum_{i=1}^{k} J_i^{|(U,X_1,..,X_n)} \delta_{Y_i}$$

where

1. *$\mu^{|U}$ is a completely random measure with intensity*

$$\nu^{|U}(ds, dx) = e^{-Us}\rho(ds|x)P_0(dx);$$

2. *$\{Y_i, i = 1, .., k\}$ are the fixed points of discontinuity, i.e. the k unique values in the sample $(X_1, \ldots, X_n)$, and the $J_i^{|(U,X1,..,X_n)}$ are the corresponding independent jumps whose density is proportional to $s^{n_i}e^{-us}\rho(ds|Y_i)$, where n_i is the number of repetitions of the value Y_i in the sample;*

3. *$\mu^{|U}$ is independent from $J_i^{|(U,X_1,..,X_n)}$, i=1,..,k.*

Note that the symbol $\stackrel{d}{=}$ stands for the equality in distribution.

Theorem 1 states that, given some latent variable U, the a-posteriori μ is still a completely random measure with fixed points of discontinuity corresponding to the locations of the observations, in particular to the positions of the unique values Y_i, i=1,..,k in the sequence $(X_1, \ldots, X_n)$.

From the previous result it is possible to derive a posterior characterization of the class of NRMI too.

Theorem 2. *If P is a NRMI with intensity $\nu(ds, dx) = \rho(ds|x)P_0(dx)$, then the posterior distribution of P given U and the data is again a NRMI with fixed points of discontinuity that coincides in distribution with the random probability measure*

$$w\frac{\mu^{|(U,X_1,..X_n)}}{T^{|(U,X_1,..X_n)}} + (1-w)\frac{\sum_{i=1}^{k} J_i^{|(U,X1,..,X_n)}\delta_{Y_i}}{\sum_{i=1}^{k} J_i^{|(U,X1,..,X_n)}},$$

where $T^{|(U,X_1,..,X_n)} = \mu^{|(U,X_1,..X_n)}(\mathbb{X})$ is the total mass of the posterior CRM and $w = T^{|(U,X_1,..X_n)}\left(T^{|(U,X_1,..X_n)} + \sum_{i=1}^{k} J_i^{|(U,X1,..,X_n)}\right)^{-1}$.

Proofs of the previous two theorems are given in James et al. (2009).

1.5 | The NGG process

In this work the focus will be on the Normalized Generalized Gamma process. As stated in Argiento et al. (2010), a generalized Gamma measure is a NRMI μ with intensity measure equal to

$$\nu(A \times B) = P_0(B) \int_A \rho(ds), \qquad A \in \mathcal{B}(\mathbb{R}^+), B \in \mathcal{B}(\mathbb{X})$$

where

$$\rho(ds) = \frac{\kappa}{\Gamma(1-\sigma)} s^{-1-\sigma} e^{-s\omega} ds, \qquad s > 0. \tag{1.6}$$

The random measure μ is homogeneous according to the decomposition of ν in Section 1.2, since $\rho(ds|x) = \rho(ds)$.

Since $\int_{\mathbb{R}^+ \times B} min(s,1)\nu(ds,dx) = P_0(B) \int_{\mathbb{R}^+} min(s,1)\rho(ds) < +\infty$, conditions (1.3) are satisfied, so that the measure is well defined. Moreover, $\int_{\mathbb{R}^+} \rho(ds) = +\infty$ guarantees the correctness of the normalization procedure, because $\mathbb{P}(0 < T := \mu(\Theta) < +\infty) = 1$. Therefore

$$P(\cdot) := \frac{\mu(\cdot)}{T} \tag{1.7}$$

defines a random probability measure on $\mathbb{X}$ which will be named *normalized generalized gamma process*, $P \sim NGG(\sigma, \kappa, \omega, P_0)$, with parameters $(\sigma, \kappa, \omega, P_0)$, where $0 \leqslant \sigma \leqslant 1$, $\omega \geq 0$, $\kappa \geq 0$. This parametrization is redundant because one degree of freedom is lost due to the normalization operation: either κ or ω can be fixed according to one's convenience. We fix $\omega = 1$ and let κ varies, changing the notation accordingly: $P \sim NGG(\sigma, \kappa, P_0)$.

Later in this chapter we will study the role assumed by the parameters, looking at the prior distribution of the number of distinct values in the sample induced by iid sampling from the process.

Since P in (1.7) is a NRMI, then

$$P = \sum_{i=1}^{+\infty} P_i \delta_{\tau_i} = \sum_{i=1}^{+\infty} \frac{J_i}{T} \delta_{\tau_i},$$

where T is the total mass of the jumps $T = \sum_i J_i$, $(J_i)_{i \geq 1}$ are the points of a Poisson process on $\mathbb{R}^+$ with mean intensity $\rho(ds)$ as in (1.6) and the locations τ_i derive from P_0. The two sequences $(\tau_i)_{i \geq 1}$ and $(J_i)_{i \geq 1}$ are independent thanks to the homogeneity of the measure μ.

It is also important to highlight that the NGG process selects discrete distributions almost surely: thus, sampling from P induces an exchangeable random partition π on the positive integers.

The first two moments of P are known in a closed analytic form for any B in $\mathcal{B}(\mathbb{X})$:

$$\mathbb{E}(P(B)) = P_0(B), \qquad Var(P(B)) = P_0(B)(1 - P_0(B))I(\sigma, \kappa)$$

where

$$I(\sigma,\kappa) := \left(\frac{1}{\sigma} - 1\right)\left(\frac{\kappa}{\sigma}\right)^{1/\sigma} e^{\left(\frac{\kappa}{\sigma}\right)}\Gamma\left(\frac{1}{\sigma}, \frac{\kappa}{\sigma}\right) = \left(\frac{1}{\sigma} - 1\right)\int_1^{+\infty} e^{-\frac{\kappa}{\sigma}(y-1)} y^{-\frac{1}{\sigma}-1} dy$$

and $\Gamma(\alpha, x) = \int_x^{+\infty} e^{-t} t^{\alpha-1} dt$ is the incomplete gamma function.

The factor $I(\sigma, \kappa)$ is decreasing as a function of each variable alone and tends to 0 when $\sigma \to 1$ or $\kappa \to +\infty$: in this case $P(B)$ converges in distribution to $P_0(B)$ for every B in $\mathcal{B}(\mathbb{X})$. On the other hand, it can be shown that

$$\lim_{\sigma \to 0, \kappa \to 0} I(\sigma, \kappa) = 1$$

so that $P(B) \xrightarrow{d} \delta_\tau(B)$, where $\tau \sim P_0$.

Within this class of priors one finds the following special cases:

1. the Dirichlet process $DP(\kappa, P_0)$ which is a $NGG(0, \kappa, P_0)$ process;
2. the normalized inverse Gaussian process that corresponds to a $NGG(1/2, \kappa, P_0)$.

1.5.1 | The prior distribution of the number of groups in a NGG process

We reveal in advance that an important issue that is addressed within mixture models is the analysis of the clustering behavior induced by the latent variables which are a sample from $P \sim NGG(\sigma, \kappa, P_0)$. Therefore the (prior) distribution of the numbers of groups in the mixture corresponds to the (prior) distribution of the number K_n of distinct observations in a sample $(X_1, \ldots, X_n)$ from a NGG process P with parameters (σ, κ, P_0). In fact, since P selects discrete distributions almost surely, there will be ties in the sample. If $k \in \{1, 2, .., n\}$ is the number of distinct values in the sample, we denote by $(X_1^*, \ldots, X_k^*)$ the distinct values and by n_i, $i = 1, \ldots, k$, the number of ties in the $i - th$ group. Obviously $\sum_{i=1}^k n_i = n$.

As proved in Lijoi et al. (2007), the prior for K_n induced by sampling from $P \sim NGG(\sigma, \kappa, P_0)$ is

$$\mathbb{P}(K_n = k) = \frac{\mathcal{G}(n, k, \sigma) exp(\beta)}{\sigma\Gamma(n)} \sum_{i=0}^{n-1} \binom{n-1}{i} (-1)^i \beta^{i/\sigma} \Gamma(k - \frac{i}{\sigma}, \beta), \;\; k = 1, \ldots, n,$$

for any $n \geqslant 1$; here, β is equal to κ/σ and

$$\mathcal{G}(n,k,\sigma) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^j \binom{k}{j}(-j\sigma)_n$$

are known as generalized Stirling numbers.

In the NGG process the parameter κ plays the same role as the mass parameter in the Dirichlet process, hence the bigger κ, the larger is the expected number of distinct values in the sample from P. In addition, also σ influences the grouping of the observations into distinct clusters: it has a double effect. On one hand, if σ increases also $\mathbb{E}(K_n)$ grows; on the other hand, σ can be used to tune the variance of the number of distinct values: the bigger σ, the flatter is the distribution of K_n. Figure 1.2 provides two examples: in the left figure we chose 3 couples of parameters (σ, κ) fixing the mean value at 6, while in the right figure the mean value is 27. It is clear that increasing σ the distribution becomes flatter, because the variance of the variable is bigger: we can deduce that a large value of σ yields a non-informative prior for the number of groups, favoring a large number of clusters with a small size.

In Figure 1.3 the mean number of K_n and its variance are represented as a function of κ (left) and σ (right): the mean and variance of K_n increase almost linearly with κ and exponentially with σ, which has a great influence for high values.

In order to choose the parameters (κ, σ) we fix the mean $\mathbb{E}(K_n)$ equal to our prior opinion on the amount of groups: however, one have to consider the constraints on the possible choices. In fact, it is possible to numerically check $\mathbb{E}_{\kappa=0,\sigma}(K_n) \leqslant \mathbb{E}_{\kappa,\sigma}(K_n)$ for any fixed σ and n (Lijoi et al., 2007).

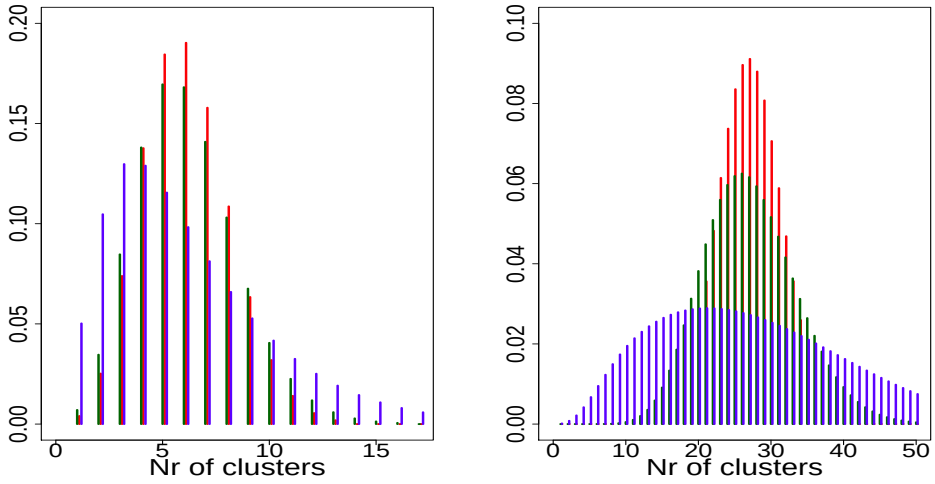


Figure 1.2: Prior distribution of the number of distinct values in a sample of $n = 150$ from a NGG(σ, κ, P_0) process.
Left: $\mathbb{E}(K_n) = 6$, $(\sigma, \kappa) = (0.002, 1.1), (0.15, 0.55), (0.3, 0.09)$ in red, dark green, blue respectively.
Right: $\mathbb{E}(K_n) = 27$, $(\sigma, \kappa) = (0.1, 7.4)$, $(0.4, 2.3)$, $(0.6, 0.2)$ in red, dark green, blue respectively.

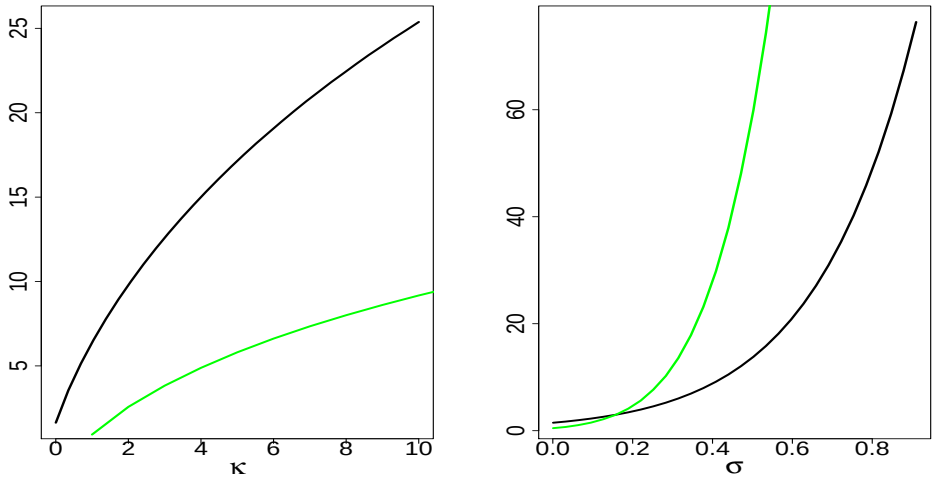


Figure 1.3: Left: Mean (black) and variance (green) of the number of clusters K_n as a function of the parameter κ, with $n = 82$, $\sigma = 0.1$.
Right: Mean (black) and variance (green) of the number of clusters K_n as a function of the parameter σ, with $n = 82$, $\kappa = 0.5$.

Chapter 2 | The ε-NGG mixture model

Until now we have focused on a class of priors based on homogeneous normalized random measures, in particular the Normalized Generalized Gamma prior. We know that a homogeneous normalized random measure can be written as a discrete measure where the weights are obtained by normalization of the jumps of a Poisson process with a fixed intensity measure, while the support is a set of iid points from a distribution P_0 on Θ:

$$P(\cdot) = \sum_{j=1}^{\infty} P_j \delta_{\tau_j}(\cdot) := \sum_{j=1}^{\infty} \left(\frac{J_j}{T}\right) \delta_{\tau_j}(\cdot). \tag{2.1}$$

The model (1.1) we are going to consider is a NRMI mixture model, i.e. a mixture model where a NRMI acts as nonparametric prior. In particular, as NRMI we chose a NGG process:

$$\begin{cases} X_i|\theta_i \overset{ind}{\sim} K(\cdot|\theta_i) & i = 1, \dots, n \\ \theta_i|P \overset{iid}{\sim} P & i = 1, \dots, n \\ P \sim NGG(\sigma, \kappa, P_0) \end{cases}$$

where K is a parametric kernel. In other words, data are assumed (conditionally) iid from a mixture of kernels $K(\cdot|\theta)$, where the mixing distribution is a normalized measure. From now on, we will consider kernels defined on $\mathbb{X} \subseteq \mathbb{R}^p$, with p an integer representing the dimension of the data, and NRMIs defined on $\Theta \subseteq \mathbb{R}^m$, the space of the parameters of the kernel. For example, if K is a univariate gaussian distribution, $N(\mu, \sigma^2)$, the latent variable θ is the vector (μ, σ^2), hence $\Theta = \mathbb{R} \times \mathbb{R}^+$. An advantage in using a NGG process mixture model instead of a DPM model is the opportunity to have an extra parameter σ, useful, together with the parameter κ, to tune the prior distribution of K_n, the number of distinct values in the sample from the process. We highlight here that using this kind of prior for a density

estimation problem, the posterior inference is made difficult by the presence of infinite unknown parameters.

We are going to solve this problem considering only jumps bigger than a threshold ε, which turns out to control the approximation to the infinite parameters prior: as we will see, conditionally to ε only a finite number of jumps has to be considered, so that the support of the mixing measure becomes finite and the computation easier. We will see in the next section that this kind of approach can be placed in the literature as an a-priori truncation method. Our prior will be called ε-NGG process of parameters $(\varepsilon, \sigma, \kappa, P_0)$. Before starting with the description of the model, we provide a short review of how this problem has been faced in the recent literature in Section 2.1.

2.1 | Some approaches in the literature

Many authors have faced the problem of the infinite mixture models when dealing with nonparametric priors. In general, there exist two main approaches: marginalization and conditional methods. The first marginalizes over the infinite dimensional process, leading to the Pólya Urn scheme in the case of mixture models: MacEachern (1998) and Neal (2000) used this approach. It has one main limit: we can not obtain information about the latent variables, since the posterior inference involves only the predictive distribution $f(X_{n+1}|X_1, \ldots, X_n)$.

On the other hand, conditional methods build a Gibbs sampler which does not integrate out the nonparametric mixing measure but updates it as a component of the algorithm itself.

Recently, Favaro and Teh (2013) developed algorithms of both types for mixture models with NRMI priors. The marginal one is a generalization of Neal's Algorithm 8 while the other one is a slice sampler, hence it adds a slice variable with a suitable distribution depending on the atoms of the prior μ. Conditioning on this variable they obtain a finite truncation for μ, having a finite number of atoms.

The reference papers on conditional algorithms for Dirichlet process mixture models are two: Papaspiliopoulos and Roberts (2008) and Walker (2007). The former built a retrospective algorithm, avoiding the need of simulating whole trajectories of the process, by inverting the order of simulation from a discrete probability. The latter proposed a slice sampler algorithm: only a finite number of jumps must be simulated, thanks to the introduction of a latent variable, conditionally on which the Dirichlet process has a representation with only a finite number of jumps. The algorithm has been extended

to NRMI mixtures in Griffin and Walker (2011).

Conditional algorithms are called *truncation* methods when the infinite parameter (i.e. the mixing measure) is approximated by truncation of the infinite sums defining the process. Truncation can be performed a-posteriori to approximate the infinite parameter P given the data as described by Gelfand and Kottas (2002) for the DPM model. On the other hand, truncation can be applied a-priori to approximate the mixing distribution with a finite dimensional random probability measure. In this way, a simpler mixture model has to be fitted. In this last framework, the pioneer works under DPM model are the ones of Ishwaran and James (2001), Ishwaran and Zarepour (2000) and Ishwaran and Zarepour (2003). For instance, Ishwaran and James (2001) consider stick breaking priors, hence random probability measures that can be built through a sequence of independent beta random variables. Their blocked Gibbs sampler uses finite approximations of the prior in order to deal with a finite number of random variables, which are updated in "blocks". As regards the NRMIs, Lijoi et al. (2007) made an analytical derivation of the posterior distribution of mixture models governed by a prior of that type: based on that result, they built a generalization of the Blackwell-Mac Queen sampling scheme which previously was suitable only for the Dirichlet process. Barrios et al. (2012) and Argiento et al. (2010) proposed a-posteriori truncation algorithms that exploit the previous characterization: the first uses the efficient inverse Lèvy measure (ILM) method developed by Wolpert and Ickstadt (1998), while the second uses the Ferguson-Klass representation of independent increment processes to update the unnormalized measure.
Recently, an a-priori truncation method has been introduced by Griffin (2013), who proposed an adaptive truncation algorithm for posterior inference in Bayesian nonparametric models involving priors both of stick-breaking and NRMI type. The level of the truncation is set by the model using a particle filters algorithm to simulate from a sequence of posterior distributions that are truncated versions of the infinite dimensional prior with an increasing number of parameters.

It is important to distinguish between two motivations for truncation. The first is studying the properties of the prior distribution, which is not our goal, and the second is posterior inference using these priors. Initial work on truncation methods was motivated by the first consideration. Concerning the Dirichlet process, the first analytical work on the approximation of the DP has been provided by Muliere and Tardella (1998): based on the stick-breaking procedure of Sethuraman (1994) and a random stopping rule, their method allows to choose in advance the degree of approximation with respect to the infinite dimensional process, guaranteeing the convergence in the Prohorov metric.

Finally, we give a motivation for using conditional algorithms: they are able to provide a *full Bayesian analysis*, i.e. it is possible to estimate the posterior mean functional (the predictive distribution), as in algorithms based on the marginalization, but also linear and non linear functionals, such as quantile functionals. We will see in Chapter 3 how to build credible intervals of the predictive distribution. In addition, these algorithms furnish the posterior chains of all the sampled variables belonging to the infinite dimensional mixture: this is not possible using methods based on marginalization, since the infinite dimensional parameter would have been integrated out.

2.2 | Construction of the P_ε prior

In order to build a finite version of (2.1) we fix $\varepsilon > 0$ and consider jumps greater than a threshold ε, as in Figure 2.1. This truncation method uses the compound Poisson process (CPP) approximation to the Lèvy measure (Griffin, 2013): in fact, the jumps larger than ε follow a compound Poisson process with intensity $\rho(x)$ for $x > \varepsilon$. In particular, the approximation considers N_ε jumps $(J_1, \dots, J_{N_\varepsilon})$ of a Poisson process with mean measure $\Lambda_\varepsilon(\cdot) = \Lambda\left((\varepsilon, +\infty) \cap \cdot\right) = \int_{(\varepsilon,+\infty)\cap\cdot} \rho(ds)$. Furthermore, the number of jumps N_ε is a random variable that follows a $Poisson\left(\Lambda\left(\varepsilon, +\infty\right)\right)$ law, so that its expectation increases as ε decreases.

Clearly $\Lambda_\varepsilon(\cdot)$ does not satisfy conditions (1.5) so that

$$\mathbb{P}\left(\sum_{j=1}^{N_\varepsilon} J_j = 0\right) > 0.$$

However, note that N_ε is almost surely finite thanks to $\Lambda_\varepsilon(\mathbb{R}^+) < +\infty$. For this reason, conditionally on N_ε, the jumps $(J_1, \dots, J_{N_\varepsilon})$ are iid from

$$\rho_\varepsilon(\cdot) = 1/\Lambda_\varepsilon(\mathbb{R}^+)\rho(\cdot)\mathbb{1}_{(\varepsilon,+\infty)}(\cdot)$$

because it turns out that $(J_1, \dots, J_{N_\varepsilon})$ is a Bernoulli process, as explained in Section 1.4. We will consider an additional point $J_0 \sim \rho_\varepsilon$ to guarantee that the total mass of the process is a.s. larger than 0, $\mathbb{P}\left(T_\varepsilon = \sum_{j=0}^{N_\varepsilon} J_j = 0\right) = 0$.

Let us consider $(\tau_0, \tau_1, \dots, \tau_{N_\varepsilon})$ iid random variables from P_0, independent of $(J_0, J_1, \dots, J_{N_\varepsilon})$. We define the following discrete random probability measure on Θ:

$$P_\varepsilon(\cdot) = \sum_{j=0}^{N_\varepsilon} P_j \delta_{\tau_j}(\cdot) := \frac{1}{T_\varepsilon} \sum_{j=0}^{N_\varepsilon} J_j \delta_{\tau_j}(\cdot), \tag{2.2}$$

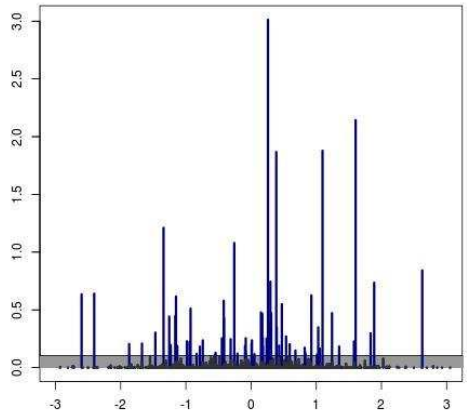


Figure 2.1: A draw $\sum_{i>0} J_i \delta_{\tau_i}$ from a homogeneous CRM on $\mathbb{R}$. Each stick denotes an atom in the CRM with mass given by its height J_i and location given by τ_i (as in Figure 1.1 (left)). Here the threshold ε is equal to 0.1: all the gray jumps (they are infinite) are discarded from the definition of the process P_ε.

identified by the triplet $\left(N_\varepsilon, (J_j)_{j=0}^{N_\varepsilon}, (\tau_j)_{j=0}^{N_\varepsilon}\right)$.

In this work we take into account in particular an ε-approximation of a normalized generalized Gamma (NGG) process. Let $\kappa > 0$, $\omega > 0$ and $0 < \sigma < 1$ be real parameters and $\varepsilon > 0$: P_ε is obtained by the normalization of $N_\varepsilon + 1$ i.i.d. random variables $J_0, \ldots, J_{N_\varepsilon}$ from $\rho_\varepsilon(\cdot)$, where $N_\varepsilon \sim$Poisson(Λ_ε),

$$\Lambda_\varepsilon := \Lambda_\varepsilon(\mathbb{R}^+) = \int_\varepsilon^{+\infty} \rho(x)dx = \frac{\kappa\omega^\sigma}{\Gamma(1-\sigma)}\Gamma(-\sigma, \omega\varepsilon),$$

and

$$\rho_\varepsilon(x) = \frac{1}{\Lambda_\varepsilon}\rho(x)\mathbb{1}_{(\varepsilon,\infty)}(x) = \frac{1}{\omega^\sigma\Gamma(-\sigma,\omega\varepsilon)}x^{-\sigma-1}\mathrm{e}^{-\omega x}\mathbb{1}_{(\varepsilon,\infty)}(x).$$

We denote by $\varepsilon-$NGG$(\sigma, \kappa, \omega, P_0)$ the law of the ε-approximation of the NGG random probability measure just defined and we will write $P_\varepsilon \sim \mathcal{P}_\varepsilon$. From equation (2.2) it follows that a ε-normalized homogeneous random probability measure, provided that P_0 is nonatomic, is a *proper* species sampling model with a random number $N_\varepsilon + 1$ of different species.

In the following Proposition 1, a result on the eppf of our ε-NGG process is provided. We see that the expression depends on an auxiliary random

variable U we already met in Section 1.4 (suggested by James et al., 2009) that we are going to reintroduce: $U|T_\varepsilon \sim Gamma(n, T_\varepsilon)$, independent on all the other variables. Clearly its density is:

$$f_{U|T_\varepsilon}(u) = \frac{1}{\Gamma(n)} T_\varepsilon^n u^{n-1} e^{-uT_\varepsilon}. \tag{2.3}$$

Integrating between 0 and $+\infty$ both members of the last formula, we obtain an alternative expression of the variable T_ε^n in terms of the integral of a function of U:

$$\frac{1}{T_\varepsilon^n} = \int_0^{+\infty} \frac{1}{\Gamma(n)} u^{n-1} e^{-uT_\varepsilon}. \tag{2.4}$$

Proposition 1. *Let $(n_1, \ldots, n_k) \subset \mathbb{N}$ be a vector such that $\sum_{i=1}^k n_i = n$. Then the eppf associated with an $P_\varepsilon \sim \varepsilon$-$NGG(\sigma, \kappa, \omega, P_0)$ is the following:*

$$\begin{aligned} p_\varepsilon(n_1, \ldots, n_k) = \int_0^\infty \frac{1}{\Gamma(n)} u^{n-1} (u+\omega)^{k\sigma - n} \prod_{i=1}^k \Gamma(n_i - \sigma, (u+\omega)\varepsilon) \\ \cdot \frac{\kappa^{k-1}}{\Gamma(1-\sigma)^{k-1}} \frac{\Lambda_{\varepsilon,u} + k}{\omega^\sigma \Gamma(-\sigma, \omega\varepsilon)} \exp\{\Lambda_{\varepsilon,u} - \Lambda_\varepsilon\} \, du, \end{aligned} \tag{2.5}$$

where $u > 0$ and

$$\Lambda_{\varepsilon,u} := \Lambda_{\varepsilon,u}(\mathbb{R}^+) = \int_\varepsilon^\infty \rho_{\varepsilon,u}(x) dx$$

with

$$\rho_{\varepsilon,u}(x) = \frac{\kappa}{\Gamma(1-\sigma)} x^{-1-\sigma} \mathrm{e}^{-(\omega+u)x} \mathbb{1}_{(0,\infty)}(x). \tag{2.6}$$

Proof. In order to prove the previous result first observe that

$$p_\varepsilon(n_1, \ldots, n_k) = \int p(n_1, \ldots, n_k | N_\varepsilon) \mathcal{L}(dN_\varepsilon). \tag{2.7}$$

Then note that thanks to the general result on the species sampling models (1.4) provided in Section 1.3, we obtain

$$p_\varepsilon(n_1, \ldots, n_k | N_\varepsilon) = \mathbb{1}_{\{1, \ldots, N_\varepsilon + 1\}}(k) \sum_{j_1, \ldots, j_k} \mathbb{E}\left(\prod_{i=1}^k P_{j_i}^{n_i} \right),$$

where the vector $(j_1, \ldots, j_k)$ ranges over all permutations of k elements in $\{0, \ldots, N_\varepsilon\}$. Therefore,

$$
\begin{aligned}
p_\varepsilon(n_1, .., n_k|N_\varepsilon) &= \mathbb{1}_{\{1,\ldots,N_\varepsilon+1\}}(k) \sum_{j_1,\ldots,j_k} \int \prod_{i=1}^{k} \frac{J_{j_i}^{n_i}}{T_\varepsilon^{n_i}} \mathcal{L}(dJ_0, \ldots, dJ_{N_\varepsilon}) \\
&= \mathbb{1}_{\{1,\ldots,N_\varepsilon+1\}}(k) \sum_{j_1,\ldots,j_k} \int \frac{1}{T_\varepsilon^n} \prod_{i=1}^{k} J_{j_i}^{n_i} \mathcal{L}(dJ_0, \ldots, dJ_{N_\varepsilon}) \\
&= \mathbb{1}_{\{1,\ldots,N_\varepsilon+1\}}(k) \sum_{j_1,\ldots,j_k} \int\int_0^\infty \frac{1}{\Gamma(n)} u^{n-1} \mathrm{e}^{-uT_\varepsilon} du \prod_{i=1}^{k} J_{j_i}^{n_i} \mathcal{L}(dJ_0, \ldots, dJ_{N_\varepsilon}) \\
&= \mathbb{1}_{\{1,\ldots,N_\varepsilon+1\}}(k) \sum_{j_1,\ldots,j_k} \int_0^\infty \Big(\frac{1}{\Gamma(n)} u^{n-1} \prod_{i=1}^{k} \int_0^\infty J_{j_i}^{n_i} \mathrm{e}^{-J_{j_i} u} \rho_\varepsilon(J_{j_i}) dJ_{j_i} \\
&\qquad \cdot \prod_{j \notin \{j_1,\ldots,j_k\}} \int_0^\infty \mathrm{e}^{-J_j u} \rho_\varepsilon(J_j) dJ_j \Big) du \\
&= \mathbb{1}_{\{1,\ldots,N_\varepsilon+1\}}(k) \sum_{j_1,\ldots,j_k} \int_0^\infty \Big(\frac{1}{\Gamma(n)} u^{n-1} \prod_{i=1}^{k} \int_\varepsilon^\infty \frac{J_{j_i}^{n_i}}{\omega^\sigma \Gamma(-\sigma, \omega\varepsilon)} J_{j_i}^{-\sigma-1} \mathrm{e}^{-(\omega+u)J_{j_i}} dJ_{j_i} \\
&\qquad \cdot \prod_{j \notin \{j_1,\ldots,j_k\}} \int_\varepsilon^\infty \frac{1}{\omega^\sigma \Gamma(-\sigma, \omega\varepsilon)} J_j^{-\sigma-1} \mathrm{e}^{-(\omega+u)J_j} dJ_j \Big) du \\
&= \mathbb{1}_{\{1,\ldots,N_\varepsilon+1\}}(k) \sum_{j_1,\ldots,j_k} \int_0^\infty \Big[\frac{1}{\Gamma(n)} u^{n-1} \prod_{i=1}^{k} \frac{(u+\omega)^{\sigma-n_i} \Gamma(n_i - \sigma; (u+\omega)\varepsilon)}{\omega^\sigma \Gamma(-\sigma, \omega\varepsilon)} \\
&\qquad \cdot \left(\frac{(u+\omega)^\sigma \Gamma(-\sigma; (u+\omega)\varepsilon)}{\omega^\sigma \Gamma(-\sigma, \omega\varepsilon)} \right)^{N_\varepsilon+1-k} \Big] du.
\end{aligned}
$$

We can change the (finite) sum with the integral. Observing that the integrand function does not depend on the position of the cluster j_i, $i = 1, \ldots, k$, but only on the numerosity n_i, we can count how many sequences of k distinct elements we can built using the elements in $\{0, \ldots, N_\varepsilon\}$. These are $(N_\varepsilon + 1)(N_\varepsilon) \ldots (N_\varepsilon + 1 - k) = \dfrac{(N_\varepsilon + 1)!}{(N_\varepsilon + 1 - k)!}$, hence

$$
\begin{aligned}
p_\varepsilon(n_1, \ldots, n_k|N_\varepsilon) &= \mathbb{1}_{\{1,\ldots,N_\varepsilon+1\}}(k) \int_0^\infty \Big[\frac{1}{\Gamma(n)} u^{n-1} \frac{(N_\varepsilon+1)!}{(N_\varepsilon+1-k)!} \\
&\cdot \prod_{i=1}^{k} \frac{(u+\omega)^{\sigma-n_i} \Gamma(n_i - \sigma; (u+\omega)\varepsilon)}{\omega^\sigma \Gamma(-\sigma, \omega\varepsilon)} \left(\frac{(u+\omega)^\sigma \Gamma(-\sigma; (u+\omega)\varepsilon)}{\omega^\sigma \Gamma(-\sigma, \omega\varepsilon)} \right)^{N_\varepsilon+1-k} \Big] du
\end{aligned}
$$

Using (2.7) and remembering N_ε has a Poisson law with parameter Λ_ε, we get

$$p_\varepsilon(n_1,\dots,n_k) = \sum_{N_\varepsilon=0}^{\infty} p_\varepsilon(n_1,\dots,n_k|N_\varepsilon)\frac{\Lambda_\varepsilon^{N_\varepsilon}}{N_\varepsilon!}\mathrm{e}^{-\Lambda_\varepsilon}.$$

By letting $N_{na} = N_\varepsilon + 1 - k$ be the number of not allocated jumps, some simple algebra gives

$$\begin{aligned} p_\varepsilon(n_1,..,n_k) \;=\; & \sum_{N_\varepsilon=0}^{\infty}\int_0^{\infty}\Big\{\frac{1}{\Gamma(n)}u^{n-1}(u+\omega)^{k\sigma-n}\prod_{i=1}^{k}\Gamma(n_i-\sigma,(u+\omega)\varepsilon) \\ & \cdot\; \frac{1}{\omega^\sigma\Gamma(-\sigma,\omega\varepsilon)}\frac{\kappa^{k-1}}{\Gamma(1-\sigma)^{k-1}} \\ & \cdot\; \frac{N_a+k}{N_a!}\left(\frac{\kappa(u+\omega)^\sigma}{\Gamma(1-\sigma)}\Gamma(-\sigma,(u+\omega)\varepsilon)\right)^{N_a}\mathrm{e}^{-\Lambda_\varepsilon}\Big\}du \end{aligned}$$

Using Fubini theorem, we exchange the integration with the series and using (2.6) to let $\Lambda_{\varepsilon,u} = \frac{\kappa(u+\omega)^\sigma}{\Gamma(1-\sigma)}\Gamma(-\sigma,(u+\omega)\varepsilon)$, we obtain

$$\sum_{N_a=0}^{\infty}\frac{N_a+k}{N_a!}\Lambda_{\varepsilon,u}^{N_a} = \mathrm{e}^{\Lambda_{\varepsilon,u}}\left(\Lambda_{\varepsilon,u}+k\right)$$

where we used the density function of a Poisson distribution and its mean. Expression (2.5) for the eppf of a $\varepsilon-\mathrm{NGG}(\sigma,\kappa,\omega,P_0)$ process follows.

□

2.3 | Weak convergence of the ε-NGG approximation

Our purpose in the previous section was to build a process that could be interpreted as a finite dimensional version of a NGG process; however, to justify our notation some convergence results are needed.

Lemma 1. *Let $\{(a_{1,n}),\dots,(a_{k,n})\}$ be a family of sequences of real numbers, with $k \geq 2$, such that*

1. $\lim_{n\to+\infty}(a_{1,n}+\cdots+a_{k,n}) = l < +\infty$,
2. $\liminf_{n\to+\infty} a_{i,n} = a_{i,0} < +\infty \quad$ *for each* $i \in \{1,\dots,k\}$,
3. $a_{1,0}+\cdots+a_{k,0} = l$.

Then

$$\lim_{n\to\infty} a_{i,n} = a_{i,0} \quad \textit{for each } i \in \{1,\dots,k\}.$$

Proof. We will prove the statement for $k = 2$, and the general result follows by induction. Since

$$\liminf_{n\to\infty}(a_{1,n}+a_{2,n}) \le \limsup_{n\to\infty} a_{1,n} + \liminf_{n\to\infty} a_{2,n} \le \limsup_{n\to\infty}(a_{1,n}+a_{2,n})$$

the following relation holds

$$l \le \limsup_{n\to\infty} a_{1,n}+a_{2,0} \le l \Rightarrow l \le \limsup_{n\to\infty} a_{1,n} + l - a_{1,0} \le l \Rightarrow \lim_{n\to\infty} a_{1,n} = a_{1,0}$$

Analogously we prove that $\lim_{n\to\infty} a_{2,n} = a_{2,0}$. □

Now we are able to show that the eppf of an ε-NGG process converges pointwise to the one of a NGG process when ε tends to 0.

Proposition 2. *Let $p_\varepsilon(\cdot)$ be the eppf of a $\varepsilon-NGG(\sigma,\kappa,\omega,P_0)$ process. Then for each $(n_1,\dots,n_k) \in \mathbb{N}$ with $k \ge 0$ and $\sum_{i=1}^k n_i = n$,*

$$\lim_{\varepsilon\to 0} p_\varepsilon(n_1,\dots,n_k) = p_0(n_1,\dots,n_k), \tag{2.8}$$

where $p_0(\cdot)$ is the eppf of a $NGG(\sigma,\kappa,\omega,P_0)$ process.

Proof. By Proposition 1

$$p_\varepsilon(n_1,\dots,n_k) = \int f_\varepsilon(u;n_1,\dots,n_k)du$$

where f_ε is the integrand in equation (2.5). Moreover the eppf of a NGG$(\sigma,\kappa,\omega,P_0)$ process can be written as

$$p_0(n_1,\dots,n_k) = \int f_0(u;n_1,\dots,n_k)du$$

where $f_0(u;n_1,..,n_k) = \frac{\kappa^k \prod_{i=1}^k \Gamma(n_i-\sigma)/\Gamma(1-\sigma)}{\Gamma(n)} u^{n-1} \exp\left\{-\kappa\frac{(\omega+u)^\sigma-\omega^\sigma}{\sigma}\right\}(u+\omega)^{k\sigma-n}$.

We first show that

$$\lim_{\varepsilon\to 0} f_\varepsilon(u;n_1,\dots,n_k) = f_0(u;n_1,\dots,n_k) \quad \forall u > 0.$$

This is straightforward by the following remarks:

1. since $(n_i - \sigma) > 0$ for each $i = 1, \ldots, k$, then
$$\lim_{\varepsilon \to 0} \Gamma(n_i - \sigma, (u + \omega)\varepsilon) = \Gamma(n_i - \sigma);$$

2. thanks to $\lim_{\varepsilon \to 0} \Gamma(-\sigma, \omega\varepsilon) = +\infty$ and the formula
$$\Gamma(1 - \sigma, x) = -\sigma\Gamma(-\sigma, x) + x^\sigma \mathrm{e}^{-x}, \tag{2.9}$$
the following relation is true for each σ and x:
$$\lim_{\varepsilon \to 0} \frac{\Lambda_{\varepsilon,u} + k}{\omega^\sigma \Gamma(-\sigma, \omega\varepsilon)} = \frac{\kappa}{\Gamma(1 - \sigma)};$$

3. by formula (2.9) and some simple analytic computations
$$\lim_{\varepsilon \to 0} \{\Lambda_{\varepsilon,u} - \Lambda_\varepsilon\} = -\kappa \frac{(\omega + u)^\sigma - \omega^\sigma}{\sigma}.$$

Now let $C = \{C_1, \ldots, C_k\}$ be a partition such that the groups numerosities are $(n_1, \ldots, n_k)$. Calling $\mathbf{p}_n$ all the possible partitions of $\{1, \ldots, n\}$ we have
$$\sum_{C_1,\ldots,C_k \in \mathbf{p}_n} p(n_1, \ldots, n_k) = 1$$
for each partition C. This holds for both p_ε and p_0.

By Fatou's Lemma we have
$$\begin{aligned}
&\int_0^\infty \liminf_{\varepsilon \to 0} f_\varepsilon(u; n_1, \ldots, n_k) du \leq \liminf_{\varepsilon \to 0} \int_0^\infty f_\varepsilon(u; n_1, \ldots, n_k) du \\
&\int_0^\infty \lim_{\varepsilon \to 0} f_\varepsilon(u; n_1, \ldots, n_k) du \leq \liminf_{\varepsilon \to 0} p_\varepsilon(n_1, \ldots, n_k) \\
&p_0(n_1, \ldots, n_k) \leq \liminf_{\varepsilon \to 0} p_\varepsilon(n_1, \ldots, n_k).
\end{aligned}$$

Suppose that for a sequence $C \in \mathbf{p}_n$, we had $p_0(n_1, .., n_k) < \liminf_{\varepsilon \to 0} p_\varepsilon(n_1, .., n_k)$. In this case
$$\begin{aligned}
1 = \sum_{C_1,\ldots,C_k \in \mathbf{p}_n} p_0(n_1, .., n_k) &< \sum_{C_1,\ldots,C_k \in \mathbf{p}_n} \liminf_{\varepsilon \to 0} p_\varepsilon(n_1, .., n_k) \leq \\
&\leq \liminf_{\varepsilon \to 0} \sum_{C_1,\ldots,C_k \in \mathbf{p}_n} p_\varepsilon(n_1, \ldots, n_k) = 1,
\end{aligned}$$
that is a contradiction. Therefore we can conclude that
$$p_0(n_1, \ldots, n_k) = \liminf_{\varepsilon \to 0} p_\varepsilon(n_1, \ldots, n_k) \quad \forall (n_1, \ldots, n_k), \ \forall k.$$
Summing up, we have proved that:

1. $\lim_{\varepsilon\to 0}\sum_{C_1,\ldots,C_k\in\mathbf{p}_n} p_\varepsilon(n_1,\ldots,n_k)=1$

2. $\liminf_{\varepsilon\to 0} p_\varepsilon(n_1,\ldots,n_k)=p_0(n_1,\ldots,n_k)\quad\forall(C_1,\ldots,C_k)\in\mathbf{p}_n$

3. $\sum_{C_1,\ldots,C_k\in\mathbf{p}_n} p_0(n_1,\ldots,n_k)=1.$

By Lemma 1, (2.8) follows. □

It is well known the law of a species sampling model is uniquely determined by the pair (P_0,p), where P_0 is a diffuse measure on Θ and p is an eppf. The next result shows how the pointwise convergence of the eppf of an $\varepsilon-NGG$ process implies the weak convergence of the corresponding random probability measure.

Proposition 3. *The law of P_ϵ converges weakly to the law of P as ϵ goes to 0, where P is a $NGG(\sigma,\kappa,\omega,P_0)$ process.*

Proof. The laws of P_ε and P are the product of two independent components, i.e. the laws of the jumps and their locations. Since the law of the locations is the same for both P_ε and P, it is sufficient to show the weak convergence of law of the jumps.

Fix $n\in\mathbb{N}$ and let $(n_{1,\varepsilon},\ldots,n_{k,\varepsilon})$ any vector of the numerosities of the clusters of a random partition $C_{n,\varepsilon}$ on $\{1,\ldots,n\}$ with eppf p_ε corresponding to an ε-NGG process. We have

$$\mathcal{L}(n_{1,\varepsilon},\ldots,n_{k,\varepsilon})\to\mathcal{L}(n_{1,0},\ldots,n_{k,0})\quad for\ \ \varepsilon\to 0, \tag{2.10}$$

where $(n_{1,0},\ldots,n_{k,0})$ is the vector of the frequencies of the clusters of a random partition $C_{n,0}$ with eppf p_0 corresponding to a NGG process. Indeed, the two laws in (2.10) are deterministic functions of the relative eppf. As n increases to infinity, by formula (134) in Pitman (2006), the random sequences $\left(\frac{n_{j,\varepsilon}}{n}\right)$ and $\left(\frac{n_{j,0}}{n}\right)$ converge to $(P_{j,\varepsilon})$ and $(P_{j,0})$ respectively, where $(P_{j,\varepsilon})$ is the sequence of the jumps of a ε-NGG process and $(P_{j,0})$ is the sequence of the jumps of a NGG process. Summarizing the previous convergence results, we have:

- $\left(\frac{n_{j,\varepsilon}}{n}\right)\xrightarrow{d}(P_{j,\varepsilon})$ for $n\to+\infty$;
- $\left(\frac{n_{j,\varepsilon}}{n}\right)\xrightarrow{d}\left(\frac{n_{j,0}}{n}\right)$ for $\varepsilon\to 0$;
- $\left(\frac{n_{j,0}}{n}\right)\xrightarrow{d}(P_{j,0})$ for $n\to+\infty$.

Recall that the weak convergence of a sequence of random probability measures is equivalent to the pointwise convergence of the Laplace functionals. Let $f(\cdot)$ be a bounded function on $\{0, 1, \dots\}$. If it is possible to invert the order of the limit operation, the following equalities prove the proposition.

$$\begin{aligned}\lim_{\varepsilon\to 0}\mathbb{E}\left(\mathrm{e}^{-\int f dP_\varepsilon}\right) &= \lim_{\varepsilon\to 0}\lim_{n\to\infty}\mathbb{E}\left(\mathrm{e}^{-\int f d\left(\frac{n_{j,\varepsilon}}{n}\right)}\right) = \lim_{n\to\infty}\lim_{\varepsilon\to 0}\mathbb{E}\left(\mathrm{e}^{-\int f d\left(\frac{n_{j,\varepsilon}}{n}\right)}\right)\\ &= \lim_{n\to\infty}\mathbb{E}\left(\mathrm{e}^{-\int f d\left(\frac{n_{j,0}}{n}\right)}\right) = \mathbb{E}\left(\mathrm{e}^{-\int f dP_0}\right).\end{aligned}$$

To justify the exchange of the two limits in the previous statement we must prove that the sequence $\mathbb{E}\left(\mathrm{e}^{-\int f d\left(\frac{n_{j,\varepsilon}}{n}\right)}\right)$ converges uniformly. To this end, it is sufficient to show that the increment from n to $(n+1)$ does not depend on ε. Calling $(n_{j,\varepsilon,s})$ the sequence of numerosities of the clusters where the total number of elements is s,

$$\begin{aligned}&\left|\mathbb{E}\left(\mathrm{e}^{-\int f d\left(\frac{n_{j,\varepsilon,n+1}}{n+1}\right)}\right) - \mathbb{E}\left(\mathrm{e}^{-\int f d\left(\frac{n_{j,\varepsilon,n}}{n}\right)}\right)\right| \leq \mathbb{E}\left(\left|\mathrm{e}^{-\int f d\left(\frac{n_{j,\varepsilon,n+1}}{n+1}\right)} - \mathrm{e}^{-\int f d\left(\frac{n_{j,\varepsilon,n}}{n}\right)}\right|\right)\\ &\leq \mathbb{E}\left(\left|\int f d\left(\frac{n_{j,\varepsilon,n+1}}{n+1}\right) - \int f d\left(\frac{n_{j,\varepsilon,n}}{n}\right)\right|\right)\end{aligned}$$

Let now $C_{n+1,\varepsilon}$ be a random partition on $\{1, \dots, n+1\}$ such that its restriction to $\{1, \dots, n\}$ corresponds to $C_{n,\varepsilon}$. We can distinguish two cases:

1. $C_{n+1,\varepsilon}$ has the same number of clusters of $C_{n,\varepsilon}$. In this case, the cluster with index j^* will have $n_j + 1$ elements;

2. $C_{n+1,\varepsilon}$ has one more cluster (of numerosity 1) than $C_{n,\varepsilon}$.

In both cases we have that

$$\mathbb{E}\left(\left|\int f d\left(\frac{n_{j,\varepsilon,n+1}}{n+1}\right) - \int f d\left(\frac{n_{j,\varepsilon,n}}{n}\right)\right|\right) \leq \frac{2M}{n+1}$$

where $M \geq \sup f$. □

2.4 | Bayesian inference for the ε-NGG mixture model

The mixture model for density estimation we consider can be hierarchically expressed as follows:

$$\begin{cases} X_1, \dots, X_n | \theta_1, \dots, \theta_n \sim \prod_{i=1}^n K(X_i|\theta_i), \\ \theta_1, \dots, \theta_n | P_\varepsilon \overset{iid}{\sim} P_\varepsilon, \\ P_\varepsilon | \varepsilon \sim \varepsilon - NGG(\sigma, \omega, \kappa, P_0) \\ \varepsilon, \sigma, \kappa \quad \sim \pi_1(\varepsilon) \cdot \pi_2(\sigma) \cdot \pi_3(\kappa) \end{cases} \tag{2.11}$$

where $K(\cdot|\theta_i)$ is a family of densities on $\mathbb{R}^p$, depending on a vector of parameters θ_i belonging to a Borel subset Θ of $\mathbb{R}^s$; P_0 is a non-atomic distribution function on Θ, expressing the "mean" of P. Model (2.11) will be addressed here as *ε-NGG hierarchical mixture model*. The Bayesian model specification is completed assuming that P_0 depends on s hyperparameters $\gamma_1, \dots, \gamma_s$ (possibly random and distributed according to $\pi(\gamma_1, \dots, \gamma_s)$).

See Chapter 3 in order to understand how the priors on σ, κ and ε affect the results, making the inference more robust.

We have constructed this kind of model starting from the well known Normalized Generalized Gamma process: we have proven in Section 2.3 that if ε decreases to 0 then the ε-NGG process tends to the NGG process. This is an interesting feature because many theoretical results about the latter process are available; for instance, we know in close form the prior distribution of the number of distinct values in the sample $(\theta_1, .., \theta_n)$ (see Section 1.5.1).

On the other hand, letting ε assume large values the ε-NGG process departs from the NGG: we are considering a different process that obviously takes into account less jumps, since more of them are cut off from the process. In particular, when ε tends to $+\infty$ the model becomes parametric. In fact, we know that $N_\varepsilon \sim Poisson(\Lambda_\varepsilon)$ for every $\varepsilon > 0$: when ε tends to $+\infty$, the parameter

$$\Lambda_\varepsilon = \int_\varepsilon^{+\infty} \frac{\kappa}{\Gamma(1-\sigma)} s^{-(1+\sigma)} e^{-\omega s} ds \to 0,$$

hence the number of components of the mixture $N_\varepsilon + 1$ tends to 1 in distribution. For that reason, the model we are considering for the data is a parametric one:

$$X_1, \dots, X_n \overset{iid}{\sim} f(x) = K(x|\tau_0),$$

where τ_0 is a draw from distribution P_0.

Unfortunately, for large values of ε we do not have many theoretical results as for NGG: more considerations about the case with a relatively large ε will be done in Chapter 3 looking at the numerical results.

We introduce some preliminary steps before starting deriving an algorithm for the posterior inference from the model (2.11). We have seen that

the prior $P_\varepsilon \sim \mathcal{P}_\varepsilon$ is identifiable from the triplet of series $\left((J_j)_{j=0}^{N_\varepsilon}, (\tau_j)_{j=0}^{N_\varepsilon}, N_\varepsilon\right)$ where the variables τs and Js are independent.

Under model (2.11), the Bayesian estimate of the real density is

$$\begin{aligned} f_{X_{n+1}}(x|X_1,..,X_n) &= \int_{\mathcal{P}\times\mathcal{R}} \left\{ \int_\Theta K(x|\theta)P(d\theta) \right\} \mathcal{L}(dP, d\varepsilon, d\sigma, d\kappa|X_1,\ldots,X_n) \\ &= \int_{\mathcal{P}\times\mathcal{R}} \sum_{j=0}^{N_\varepsilon} \frac{J_j}{T_\varepsilon} K(x|\tau_j)\mathcal{L}(dP, d\varepsilon, d\sigma, d\kappa|X_1,\ldots,X_n), \end{aligned}$$

where $\mathcal{R} = \mathbb{R}^+ \times (0,1) \times \mathbb{R}^+$. This integral, as well as the other posterior inferences, must be computed via a MCMC algorithm. With this aim, we sample a Markovian sequence $\{\varepsilon^{(b)}, \sigma^{(b)}, \kappa^{(b)}, P_\varepsilon^{(b)}\}_{b=1}^B$ from the posterior law $\mathcal{L}(d\varepsilon, d\sigma, d\kappa, dP_\varepsilon|X_1,\ldots,X_n)$ with B large enough. The density estimation then becomes

$$f_{X_{n+1}}(x|X_1,..,X_n) \simeq \frac{1}{B}\sum_{b=1}^{B}\sum_{j=0}^{N_\varepsilon^{(b)}} \frac{J_j^{(b)}}{T_\varepsilon^{(b)}} K(x|\tau_j^{(b)}).$$

If we enlarge the state space by $\theta = (\theta_1,\ldots,\theta_n)$ and by the auxiliary variable U, we can build a Gibbs sampler algorithm.

First of all we provide some notation issues and then a result concerning the posterior distribution. Let $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_n)$ be a sample from P_ε and set the variable $U := \Gamma_n/T_\varepsilon$, where $\Gamma_n \sim gamma(n,1)$. The following proposition gives a "finite dimensional" version of the characterization of the posterior law of a NGG process in James et al. (2009). As in the infinite dimensional case, the posterior distribution of an ε-NGG$(\sigma,\kappa,\omega,P_0)$ process, conditionally on U and $\boldsymbol{\theta}$, can be expressed as the law of a random probability measure, which is a mixture between a ε-NGG process and a discrete probability measure with support given by the (observed) distinct values $\boldsymbol{\theta}^* = (\theta_1^*,\ldots,\theta_k^*)$. We will call *allocated jumps* of the process the values $J_{l_1^*}, J_{l_2^*},\ldots,J_{l_k^*}$ such that there exists a corresponding location for which $\tau_{l_i^*} = \theta_i^*$, $i = 1,\ldots,k$. The other values will be called *non-allocated jumps*.

Proposition 4. *If P_ε is a $\varepsilon-NGG(\sigma,\kappa,\omega,P_0)$ process, then the posterior distribution of P_ε coincides with that of the random measure*

$$P_\varepsilon^*(\cdot) = wP_{\varepsilon,u}^{(na)}(\cdot) + (1-w)\sum_{j=1}^{k} P_j^{(a)}\delta_{\theta_k^*}(\cdot)$$

where U is the variable defined in (2.3) *and*

1. *$P^{(na)}_{\varepsilon,u}(\cdot)$, the process of not assigned jumps, is distributed according to an $\varepsilon-NGG(\sigma,\kappa,\omega+u,P_0)$ process, conditional to have $N^{(na)}$ jumps, where*
$$N^{(na)} \sim \frac{\Lambda_{\varepsilon,u}}{k+\Lambda_{\varepsilon,u}}\left(\mathcal{P}(\Lambda_{\varepsilon,u})+1\right)+\frac{k}{k+\Lambda_{\varepsilon,u}}\mathcal{P}(\Lambda_{\varepsilon,u})$$
with $\mathcal{P}$ a Poisson density.

2. *The jumps $\{P^{(a)}_1,\dots,P^{(a)}_k\}$ assigned to the fixed points of discontinuity $\boldsymbol{\theta}^*=(\theta^*_1,\dots,\theta^*_k)$ of P^*_ε are obtained by normalization of*
$$J^{(a)}_j \sim gamma(n_j-\sigma, u+\omega),\ for\ j=1\dots,k.$$

3. *$P^{(na)}_{\varepsilon,u}(\cdot)$ and $\{J^{(a)}_1,\cdots,J^{(a)}_k\}$ are independent.*

Moreover, if $N^{(na)}$ is different from 0, $w=T_{\varepsilon,u}/(T_{\varepsilon,u}+\sum_{j=1}^k J^{(a)}_j)$ where $T_{\varepsilon,u}$ is the normalization variable in the representation of $P^{(na)}_{\varepsilon,u}(\cdot)$ as in (2.2).

This result will be clarified later, when in the Gibbs sampler we will obtain the full-conditional for P_ε : now we begin to derive a method to sample from the posterior distribution. Note that

$$\begin{aligned}\mathcal{L}(\theta_1,..,\theta_n|P_\varepsilon)&=\prod_{i=1}^n P_\varepsilon(\theta_i)=\prod_{i=1}^n\sum_{j=0}^{N_\varepsilon}\left(P_j\delta_{\tau_j}(\theta_i)\right)\\&=\sum_{l_1=0}^{N_\varepsilon}P_{l_1}\delta_{\tau_{l_1}}(\theta_1)\sum_{l_2=0}^{N_\varepsilon}P_{l_2}\delta_{\tau_{l_2}}(\theta_2)\dots\sum_{l_n=0}^{N_\varepsilon}P_{l_n}\delta_{\tau_{l_n}}(\theta_n)\\&=\sum_{l^*_1,\dots,l^*_k}\left(P^{n_1}_{l^*_1}\dots P^{n_k}_{l^*_k}\delta_{\tau_{l^*_1}}(\theta^*_1)\dots\delta_{\tau_{l^*_k}}(\theta^*_k)\right)\end{aligned}\tag{2.12}$$

where $(\theta^*_1,\theta^*_2,\dots,\theta^*_k)$ represents the vector of the unique values of the sample $\boldsymbol{\theta}$: any value is repeated n_i times. This induces a partition over the indices of the data into k groups $\{C_1,C_2,\dots,C_k\}$ that will be useful later. Moreover, the law of the data and the latent variables θ_i, $i=1,..,n$, conditional to P_ε is the following:

$$\begin{aligned}\mathcal{L}(\mathbf{X},\boldsymbol{\theta}|P_\varepsilon)&=\prod_{i=1}^n\left(P_\varepsilon(\theta_i)K(X_i|\theta_i)\right)\\&\overset{(2.12)}{=}\Big(\prod_{i\in C_1}K(X_i|\theta^*_1)..\prod_{i\in C_k}K(X_i|\theta^*_k)\Big)\sum_{l^*_1,..,l^*_k}\left(P^{n_1}_{l^*_1}..P^{n_k}_{l^*_k}\delta_{\tau_{l^*_1}}(\theta^*_1)..\delta_{\tau_{l^*_k}}(\theta^*_k)\right)\\&=\frac{1}{T^n_\varepsilon}\sum_{l^*_1,..,l^*_k}\Big(J^{n_1}_{l^*_1}\prod_{i\in C_1}K(X_i|\theta^*_1)J^{n_2}_{l^*_2}\prod_{i\in C_2}K(X_i|\theta^*_2)..J^{n_k}_{l^*_k}\prod_{i\in C_k}K(X_i|\theta^*_k)\Big).\end{aligned}\tag{2.13}$$

Building our Gibbs sampler we will consider the law $\mathcal{L}(\mathbf{X}, \boldsymbol{\theta}|P_\varepsilon)$ as the marginal of $\mathcal{L}(\mathbf{X}, \boldsymbol{\theta}, u|P_\varepsilon)$. In particular, thanks to the independence of the variables, the expressions (2.4) and (2.13), we obtain

$$\mathcal{L}(\mathbf{X}, \boldsymbol{\theta}, u|P_\varepsilon) = \frac{1}{\Gamma(n)} u^{n-1} e^{-uT_\varepsilon} \sum_{l_1^*,..,l_k^*} \big(J_{l_1^*}^{n_1} \prod_{i \in C_1} K(X_i|\theta_1^*) \dots J_{l_k^*}^{n_k} \prod_{i \in C_k} K(X_i|\theta_k^*) \big)$$

Therefore, the conjugate law of the entire model can be written in the following way:

$$\begin{aligned}
&\mathcal{L}(\mathbf{X}, \boldsymbol{\theta}, u, P_\varepsilon, \varepsilon) = \mathcal{L}(\mathbf{X}, \boldsymbol{\theta}, u|P_\varepsilon, \varepsilon)\mathcal{L}(P_\varepsilon, \varepsilon) = \mathcal{L}(\mathbf{X}, \boldsymbol{\theta}, u|P_\varepsilon, \varepsilon)\mathcal{L}(P_\varepsilon|\varepsilon)\mathcal{L}(\varepsilon) \\
&= \frac{1}{\Gamma(n)} u^{n-1} e^{-uT_\varepsilon} \sum_{l_1^*,..,l_k^*} \big(J_{l_1^*}^{n_1} \prod_{i \in C_1} K(X_i|\theta_1^*) .. J_{l_k^*}^{n_k} \prod_{i \in C_k} K(X_i|\theta_k^*) \big) \mathcal{L}(P_\varepsilon|\varepsilon)\pi(\varepsilon) \\
&= \frac{1}{\Gamma(n)} u^{n-1} \prod_{j=0}^{N_\varepsilon} (e^{-uJ_j}) \sum_{l_1^*,...,l_k^*} \big(J_{l_1^*}^{n_1} \prod_{i \in C_1} K(X_i|\theta_1^*) .. J_{l_k^*}^{n_k} \prod_{i \in C_k} K(X_i|\theta_k^*) \big) \cdot \\
&\qquad \cdot \prod_{j=0}^{N_\varepsilon} \big(\rho_\varepsilon(J_j) P_0(\tau_j) \big) \mathcal{P}(N_\varepsilon|\Lambda_\varepsilon)\pi(\varepsilon) \\
&= \frac{1}{\Gamma(n)} u^{n-1} \prod_{j=0}^{N_\varepsilon} \big(e^{-uJ_j} \rho_\varepsilon(J_j) P_0(\tau_j) \big) \sum_{l_1^*,..,l_k^*} \big(J_{l_1^*}^{n_1} \prod_{i \in C_1} K(X_i|\theta_1^*) .. J_{l_k^*}^{n_k} \prod_{i \in C_k} K(X_i|\theta_k^*) \big) \\
&\qquad \cdot \mathcal{P}(N_\varepsilon|\Lambda_\varepsilon)\pi(\varepsilon)
\end{aligned} \tag{2.14}$$

where $\mathcal{P}(N_\varepsilon|\Lambda_\varepsilon)$ is the density of the random variable N_ε distributed as a Poisson of parameter Λ_ε. In the previous relation we used the characterization of P_ε in form of infinite summation in order to write the law of P_ε given ε as the law of the vectors $\mathbf{J}, \boldsymbol{\tau}$ and N_ε. Moreover,

$$\begin{aligned}
\mathcal{L}(P_\varepsilon|\varepsilon) &= \mathcal{L}((J_j)_{j=0}^{N_\varepsilon}, (\tau_j)_{j=0}^{N_\varepsilon}, N_\varepsilon|\varepsilon) = \mathcal{L}(\mathbf{J}|N_\varepsilon)\mathcal{L}(\boldsymbol{\tau}|N_\varepsilon)\mathcal{L}(N_\varepsilon|\varepsilon) \\
&= \mathcal{P}(N_\varepsilon|\Lambda_\varepsilon) \prod_{j=0}^{N_\varepsilon} \left(\rho_\varepsilon(J_j) P_0(\tau_j) \right).
\end{aligned}$$

The factorization is due to the independence between the vectors $\mathbf{J}$ and $\boldsymbol{\tau}$. The product arises from the fact that the τ_i are iid from $P_0(\cdot)$ on the state space Θ while the J_i are iid from $\rho_\varepsilon(\cdot)$ on $\mathbb{R}^+$.

In order to construct a Gibbs sampler to sample from the posterior of P_ε, we are going to list and describe every updating step of the algorithm, namely

the full-conditionals. We highlight that in the previous (and the following) passages only the parameter ε is been considered as a random variable: later, we will introduce updating steps in the algorithm for the cases where also σ and κ are random.

2.4.1 | Gibbs Sampler

Update U

Since the variable u is independent from the others, the posterior law will be again

1. $$\mathcal{L}(u|\mathbf{X}, \boldsymbol{\theta}, P_\varepsilon) = Gamma(n, T_\varepsilon)$$

where T_ε is the sum of all the masses.

Update $\boldsymbol{\theta}$

Thanks to the Bayes theorem we have

$$\mathcal{L}(\boldsymbol{\theta}|u, \mathbf{X}, P_\varepsilon) \propto \mathcal{L}(\mathbf{X}, u, \boldsymbol{\theta}, P_\varepsilon),$$

so from the relation (2.14), omitting all the parts not depending on $\boldsymbol{\theta}$, we obtain a discrete law with support on all the τ_j, for every $i = 1, \dots n$,

2. $$\mathbb{P}(\theta_i = \tau_j) \propto J_j K(X_i|\tau_j), \;\; j = 0, .., N_\varepsilon.$$

Update $\boldsymbol{P_\varepsilon}$

This is the most complicated step, because the law P_ε is a combination of different contributions, the jumps $\mathbf{J}$, the locations $\boldsymbol{\tau}$, the number of components of the mixture N_ε and the level of approximation ε. Observe now that

$$\begin{aligned}
&\mathcal{L}(P_\varepsilon|u, \boldsymbol{\theta}, \mathbf{X}) = \mathcal{L}(\varepsilon, N_\varepsilon, \mathbf{J}, \boldsymbol{\tau}|u, \boldsymbol{\theta}, \mathbf{X}) \propto \\
&\mathcal{L}(\mathbf{J}, \boldsymbol{\tau}|u, \boldsymbol{\theta}, \mathbf{X}, \varepsilon, N_\varepsilon)\mathcal{L}(\varepsilon, N_\varepsilon|u, \boldsymbol{\theta}, \mathbf{X}) \propto \\
&\mathcal{L}(\mathbf{J}, \boldsymbol{\tau}|u, \boldsymbol{\theta}, \mathbf{X}, \varepsilon, N_\varepsilon)\mathcal{L}(N_\varepsilon|\varepsilon, u, \boldsymbol{\theta}, \mathbf{X})\mathcal{L}(\varepsilon|u, \boldsymbol{\theta}, \mathbf{X}).
\end{aligned}$$

This relation suggests a way to hierarchically sample from the posterior $\mathcal{L}(P_\varepsilon|u,\boldsymbol{\theta},\mathbf{X})$: first sample ε from $\mathcal{L}(\varepsilon|u,\boldsymbol{\theta},\mathbf{X})$, then N_ε from $\mathcal{L}(N_\varepsilon|\varepsilon,u,\boldsymbol{\theta},\mathbf{X})$ and finally the jumps and the points of support from $\mathcal{L}(\mathbf{J},\boldsymbol{\tau}|u,\boldsymbol{\theta},\mathbf{X},\varepsilon,N_\varepsilon)$. Proceeding with the updating step of N_ε and ε, we obtain:

$$
\begin{aligned}
&\mathcal{L}(N_\varepsilon,\varepsilon|u,\boldsymbol{\theta},\mathbf{X}) \propto \mathcal{L}(N_\varepsilon,\varepsilon,u,\boldsymbol{\theta},\mathbf{X}) = \int \mathcal{L}(N_\varepsilon,\varepsilon,u,\boldsymbol{\theta},\mathbf{J},\boldsymbol{\tau})d\mathbf{J}d\boldsymbol{\tau} \\
&= \int \Bigg[\frac{1}{\Gamma(n)} u^{n-1} \prod_{j=0}^{N_\varepsilon} \left(e^{-uJ_j}\rho_\varepsilon(J_j)P_0(\tau_j)\right) \sum_{l_1^*,\dots,l_k^*} \Big(J_{l_1^*}^{n_1} \prod_{i\in C_1} K(X_i|\theta_1^*) \\
&\qquad\qquad \dots J_{l_k^*}^{n_k} \prod_{i\in C_k} K(X_i|\theta_k^*) \Big) \mathcal{P}(N_\varepsilon|\Lambda_\varepsilon)\pi(\varepsilon) \Bigg] d\mathbf{J}d\boldsymbol{\tau} \\
&= \sum_{l_1^*,\dots,l_k^*} \int \Bigg[\prod_{i=1}^{k} \Big(J_{l_i^*}^{n_i} \prod_{j\in C_i} K(X_j|\theta_i^*) \Big) \prod_{j=0}^{N_\varepsilon} \Big(e^{-uJ_j}\rho_\varepsilon(J_j)P_0(\tau_j) \Big) \Bigg] d\mathbf{J}d\boldsymbol{\tau} \qquad (2.15) \\
&\qquad\qquad \cdot \left(\frac{1}{\Gamma(n)} u^{n-1} \mathcal{P}(N_\varepsilon|\Lambda_\varepsilon)\pi(\varepsilon) \right) \\
&= \sum_{l_1^*,\dots,l_k^*} \Bigg\{ \Bigg[\prod_{i=1}^{k} \int J_{l_i^*}^{n_i} \prod_{j\in C_i} K(X_j|\theta_i^*) e^{-uJ_{l_i^*}} \rho_\varepsilon(J_{l_i^*})P_0(\tau_{l_i^*}) dJ_{l_i^*} d\tau_{l_i^*} \Bigg] \\
&\cdot \Bigg[\prod_{j\neq\{l_1^*,..,l_k^*\}}^{N_\varepsilon} \int e^{-uJ_j}\rho_\varepsilon(J_j)P_0(\tau_j)dJ_j d\tau_j \Bigg] \Bigg\} \frac{1}{\Gamma(n)} u^{n-1}\mathcal{P}(N_\varepsilon|\Lambda_\varepsilon)\pi(\varepsilon).
\end{aligned}
$$

We do not specify the domain of the integrals for simplicity of notation. Observe that the integral in the second parenthesis for the non-allocated jumps is equal to

$$
\begin{aligned}
&\int e^{-uJ_j}\rho_\varepsilon(J_j)P_0(\tau_j)dJ_j d\tau_j = \int e^{-uJ_j}\rho_\varepsilon(J_j)dJ_j \\
&= \frac{(\omega+u)^\sigma}{\omega^\sigma\Gamma(-\sigma,\omega\varepsilon)} \int_{(\omega+u)\varepsilon}^{\infty} e^{-y}y^{-\sigma-1}dy = \frac{(\omega+u)^\sigma\Gamma(-\sigma,(\omega+u)\varepsilon)}{\omega^\sigma\Gamma(-\sigma,\omega\varepsilon)}.
\end{aligned}
$$

On the other hand, the first squared parenthesis in expression (2.15) , thanks to the relation $\tau_{l_i^*} = \theta_i^*$, becomes:

$$
\begin{aligned}
&\prod_{i=1}^{k} \int J_{l_i^*}^{n_i} \prod_{j\in C_i} K(X_j|\theta_i^*) e^{-uJ_{l_i^*}} \rho_\varepsilon(J_{l_i^*})P_0(\tau_{l_i^*}) dJ_{l_i^*} d\tau_{l_i^*} \\
&= \prod_{i=1}^{k} \int J_{l_i^*} \prod_{j\in C_i} K(X_j|\theta_i^*) e^{-uJ_{l_i^*}} \rho_\varepsilon(J_{l_i^*})P_0(\theta_i^*) dJ_{l_i^*} d\theta_i^* \qquad (2.16) \\
&= \prod_{i=1}^{k} \left(\int J_{l_i^*} e^{-uJ_{l_i^*}} \rho_\varepsilon(J_{l_i^*}) dJ_{l_i^*} \right) \left(\int \prod_{j\in C_i} K(X_j|\theta_i^*)P_0(\theta_i^*)d\theta_i^* \right).
\end{aligned}
$$

The last parenthesis will be named $m(C_i) := \int \prod_{j \in C_i} K(X_j|\theta_i^*) P_0(\theta_i^*) d\theta_i^*$; it is the marginal law of the data in the i-th group.

Going on with the computation of the part of the expression (2.16) associated with the allocated jumps, i.e. the first parenthesis, we have:

$$\int J_{l_i^*} e^{-uJ_{l_i^*}} \rho_\varepsilon(J_{l_i^*}) dJ_{l_i^*} = \frac{1}{\omega^\sigma \Gamma(-\sigma, \omega\varepsilon)} \int_\varepsilon^\infty x^{n_i} e^{-ux} x^{-1-\sigma} e^{-\omega x} dx$$
$$= \frac{(\omega+u)^{\sigma-n_i}}{\omega^\sigma} \frac{\Gamma(n_i - \sigma, (u+\omega)\varepsilon)}{\Gamma(-\sigma, \omega\varepsilon)},$$

where we used the change of variable in the integral $(u+\omega)x = y$ (as before) and the definition of incomplete gamma function.

Expression (2.15), that represents the updating part of the paramater ε and the number of components of the mixture N_ε, then becomes the following:

$$\frac{1}{\Gamma(n)} u^{n-1} \sum_{l_1^*,..,l_k^*} \left\{ \left(\frac{(\omega+u)^{k\sigma-n} \prod_{i=1}^k \Gamma(n_i - \sigma, (\omega+u)\varepsilon) m(C_i)}{\omega^{\sigma k} \Gamma(-\sigma, \omega\varepsilon)^k} \right) \right.$$
$$\left. \cdot \left(\frac{(\omega+u)^{\sigma(N_\varepsilon+1-k)} \Gamma(-\sigma, (u+\omega)\varepsilon)^{N_\varepsilon+1-k}}{\omega^{\sigma(N_\varepsilon+1-k)} \Gamma(-\sigma, \omega\varepsilon)^{N_\varepsilon+1-k}} \right) \right\} \mathcal{P}(N_\varepsilon|\Lambda_\varepsilon) \pi(\varepsilon)$$
$$= \frac{u^{n-1}}{\Gamma(n)} \mathcal{P}(N_\varepsilon|\Lambda_\varepsilon) \pi(\varepsilon) \frac{(N_\varepsilon+1)!}{(N_\varepsilon+1-k)!} \prod_{i=1}^k \left(m(C_i) \Gamma(n_i - \sigma, \varepsilon(\omega+u)) \right)$$
$$\cdot \frac{(\omega+u)^{\sigma k-n}}{\omega^{\sigma k} \Gamma(-\sigma, \omega\varepsilon)^k} \frac{(\omega+u)^{\sigma N_{na}} \Gamma(-\sigma, \varepsilon(\omega+u))^{N_{na}}}{\omega^{\sigma N_{na}} \Gamma(-\sigma, \omega\varepsilon)^{N_{na}}} \mathbb{1}_{\{(N_\varepsilon+1) \geq k\}}$$

In the previous computation, we have exploited the fact that no terms in the summation depend explicitly on $l_1^*, \ldots, l_k^*$: we replaced the summation with the number of all possible combination of indices, that is $\frac{(N_\varepsilon+1)!}{(N_\varepsilon+1-k)!}$ with $N_\varepsilon + 1 \geq k$. Moreover, we denoted by $N_{na} = N_\varepsilon + 1 - k$ the number of non-allocated jumps.

After these observations, the law (2.15) becomes the following:

$$\begin{aligned}
\mathcal{L}(N_\varepsilon, \varepsilon | u, \boldsymbol{\theta}, \mathbf{X}) &\propto \frac{u^{n-1}}{\Gamma(n) N_\varepsilon!} e^{-\Lambda_\varepsilon} \Lambda_\varepsilon^{N_\varepsilon} \pi(\varepsilon) \mathbb{1}_{(N_\varepsilon+1 \geq k)} \frac{(N_\varepsilon+1)!}{N_{na}!} \\
&\quad \cdot \frac{(\omega+u)^{\sigma(N_\varepsilon+1)-n} \Gamma(-\sigma, \varepsilon(\omega+u))^{N_{na}}}{\omega^{\sigma(N_\varepsilon+1)} \Gamma(-\sigma, \omega\varepsilon)^{N_\varepsilon+1}} \prod_{i=1}^{k} \Big(m(C_i) \Gamma(n_i - \sigma, \varepsilon(u+\omega)) \Big) \\
&= \frac{u^{n-1}}{\Gamma(n)} \frac{N_\varepsilon+1}{N_{na}!} e^{-\Lambda_\varepsilon} \Big(\frac{\kappa \omega^\sigma}{\Gamma(1-\sigma)} \Gamma(-\sigma, \omega\varepsilon) \Big)^{N_\varepsilon} \pi(\varepsilon) \mathbb{1}_{(N_\varepsilon+1 \geq k)} \\
&\quad \cdot \frac{(\omega+u)^{\sigma(N_\varepsilon+1)-n} \Gamma(-\sigma, \varepsilon(\omega+u))^{N_{na}}}{\omega^{\sigma(N_\varepsilon+1)} \Gamma(-\sigma, \omega\varepsilon)^{N_\varepsilon+1}} \prod_{i=1}^{k} \Big(m(C_i) \Gamma(n_i - \sigma, \varepsilon(u+\omega)) \Big) \\
&= \frac{u^{n-1}}{\Gamma(n)} \Big(\frac{\kappa}{\Gamma(1-\sigma)} \Big)^{k-1} \pi(\varepsilon) \mathbb{1}_{(N_{na} \geq 0)} e^{-\Lambda_\varepsilon} \prod_{i=1}^{k} \Big(m(C_i) \Gamma(n_i - \sigma, \varepsilon(u+\omega)) \Big) \\
&\qquad \cdot \frac{(\omega+u)^{\sigma k - n}}{\omega^\sigma \Gamma(-\sigma, \omega\varepsilon)} \frac{(N_{na}+k)}{N_{na}!} \Big(\frac{\kappa (u+\omega)^\sigma}{\Gamma(1-\sigma)} \Gamma(-\sigma, (u+\omega)\varepsilon) \Big)^{N_{na}}
\end{aligned}$$

where we remembered the equality $\Lambda_\varepsilon = \dfrac{\kappa \omega^\sigma}{\Gamma(1-\sigma)} \Gamma(-\sigma, \omega\varepsilon)$.

If we call $\rho_{\varepsilon,u}(x) = \rho_\varepsilon(x) e^{-ux} \omega^\sigma = \dfrac{\kappa}{\Gamma(1-\sigma)} x^{-1-\sigma} e^{-x(u+\omega)} \mathbb{1}_{(\varepsilon, +\infty)}(x)$, then the total mass of this new function $\rho_{\varepsilon,u}$ is

$$\Lambda_{\varepsilon,u} = \Lambda_{\varepsilon,u}(\mathbb{R}^+) = \int_0^{+\infty} \rho_{\varepsilon,u}(x) dx = \frac{\kappa (u+\omega)^\sigma}{\Gamma(1-\sigma)} \Gamma(-\sigma, \varepsilon(u+\omega)).$$

At the end, expression (2.15) is equal to

$$\begin{aligned}
\mathcal{L}(N_\varepsilon, \varepsilon | u, \boldsymbol{\theta}, \mathbf{X}) &\propto \frac{u^{n-1}}{\Gamma(n)} \Big(\frac{\kappa}{\Gamma(1-\sigma)} \Big)^{k-1} \pi(\varepsilon) \prod_{i=1}^{k} \Big[m(C_i) \Gamma(n_i - \sigma, \varepsilon(u+\omega)) \Big] \\
&\qquad \cdot \frac{(\omega+u)^{\sigma k - n}}{\omega^\sigma \Gamma(-\sigma, \omega\varepsilon)} (N_{na}+k) e^{-\Lambda_\varepsilon} \frac{1}{N_{na}!} (\Lambda_{\varepsilon,u})^{N_{na}} \mathbb{1}_{(N_{na} \geq 0)} e^{-\Lambda_{\varepsilon,u}} e^{\Lambda_{\varepsilon,u}} \\
&= \frac{u^{n-1}}{\Gamma(n)} \Big(\frac{\kappa}{\Gamma(1-\sigma)} \Big)^{k-1} \prod_{i=1}^{k} \Big[m(C_i) \Gamma(n_i - \sigma, \varepsilon(u+\omega)) \Big] \\
&\qquad \cdot \frac{(\omega+u)^{\sigma k - n}}{\omega^\sigma \Gamma(-\sigma, \omega\varepsilon)} (N_{na}+k) \cdot \mathcal{P}(N_{na} | \Lambda_{\varepsilon,u}) e^{\Lambda_{\varepsilon,u} - \Lambda_\varepsilon} \pi(\varepsilon).
\end{aligned}$$

Observing the previous relation it is clear that the posterior law of the number

of non-allocated jumps, N_{na}, is a mixture of two Poisson's laws:

$$\begin{aligned}\mathcal{L}(N_{na}|\varepsilon,u,\boldsymbol{\theta},\mathbf{X}) &\propto \frac{N_{na}+k}{N_{na}!}e^{-\Lambda_{\varepsilon,u}}\Lambda_{\varepsilon,u}^{N_{na}} = \left(\frac{N_{na}}{N_{na}!}+\frac{k}{N_{na}!}\right)e^{-\Lambda_{\varepsilon,u}}\Lambda_{\varepsilon,u}^{N_{na}} \\ &= \frac{\Lambda_{\varepsilon,u}}{(N_{na}-1)!}\Lambda_{\varepsilon,u}^{(N_{na}-1)}e^{-\Lambda_{\varepsilon,u}} + \frac{k}{N_{na}!}\Lambda_{\varepsilon,u}^{N_{na}}e^{-\Lambda_{\varepsilon,u}} \\ &\propto \frac{\Lambda_{\varepsilon,u}}{\Lambda_{\varepsilon,u}+k}\mathcal{P}(N_{na}^{*}|\Lambda_{\varepsilon,u}) + \frac{k}{\Lambda_{\varepsilon,u}+k}\mathcal{P}(N_{na}|\Lambda_{\varepsilon,u})\end{aligned} \tag{2.17}$$

where $N_{na}^{*} = N_{na} - 1$.

Therefore the law of N_{ε} is that of $N_{na} + k - 1$, where N_{na} is a sample from a mixture of two Poissons with parameter $\Lambda_{\varepsilon,u}$.

Moreover, the posterior law of the approximation parameter ε is:

$$\begin{aligned}\mathcal{L}(\varepsilon|u,\boldsymbol{\theta},\mathbf{X}) &\propto \sum_{N_{na}=0}^{+\infty}\mathcal{L}(N_{na},\varepsilon|u,\boldsymbol{\theta},\mathbf{X}) \\ &= \frac{u^{n-1}}{\Gamma(n)}\left(\frac{\kappa}{\Gamma(1-\sigma)}\right)^{k-1}\prod_{i=1}^{k}\Big[m(C_i)\Gamma(n_i-\sigma,\varepsilon(u+\omega))\Big]\pi(\varepsilon) \\ &\qquad \cdot\frac{(\omega+u)^{\sigma k-n}}{\omega^{\sigma}\Gamma(-\sigma,\omega\varepsilon)}e^{\Lambda_{\varepsilon,u}-\Lambda_{\varepsilon}}\sum_{N_{na}=0}^{+\infty}(N_{na}+k)\mathcal{P}(N_{na}|\Lambda_{\varepsilon,u}) \\ &\propto \prod_{i=1}^{k}\Gamma(n_i-\sigma,\varepsilon(u+\omega))(\omega+u)^{\sigma k-n}e^{(\Lambda_{\varepsilon,u}-\Lambda_{\varepsilon})}\frac{\Lambda_{\varepsilon,u}+k}{\Gamma(-\sigma,\omega\varepsilon)}\pi(\varepsilon)\end{aligned} \tag{2.18}$$

At the end, the scheme of this step of the Gibbs sampler can be summarized in the following way:

3. Sample ε from law (2.18);

4. Sample N_{na} from the mixture in (2.17) and compute the number of components in the mixture as $N_{\varepsilon} + 1 = N_{na} + k$;

5. Sample the $N_{\varepsilon} + 1$ jumps, $(J_0, J_1, \ldots, J_{N_{\varepsilon}})$.

If the jump is non-allocated, then sample from the following density (compare expression (2.14)):

$$\mathcal{L}(J_j|N_{\varepsilon},\varepsilon,u,\mathbf{X},\boldsymbol{\theta}) \propto \mathcal{L}(J_j,N_{\varepsilon},\varepsilon,u,\mathbf{X},\boldsymbol{\theta}) \propto e^{-uJ_j}\rho_{\varepsilon}(J_j) = \rho_{u,\varepsilon}(J_j) \tag{2.19}$$

On the other hand, if the jump is allocated, so there exists an index l_i^* such that $\theta_i = \tau_{l_i^*}$, we have to independently sample from:

$$\mathcal{L}(J_{l_i^*}|N_\varepsilon, \varepsilon, u, \mathbf{X}, \boldsymbol{\theta}) \propto J_{l_i^*}^{n_i} e^{-uJ_{l_i^*}} \rho_\varepsilon(J_{l_i^*}) \propto Gamma(n_i - \sigma, \omega + u)\mathbb{1}_{(\varepsilon,+\infty)}$$

6. Sample the $N_\varepsilon + 1$ points of the support, $(\tau_0, \tau_1, \ldots, \tau_{N_\varepsilon})$. As for the jumps, for the non-allocated points we have simply to sample i.i.d. from P_0; instead, for the allocated ones, the sample is i.i.d. from:

$$\mathcal{L}(\theta_i^*|N_\varepsilon, \varepsilon, u, \mathbf{X}, \boldsymbol{\theta}) \propto \left\{ \prod_{j \in C_i} K(X_j|\theta_i^*) \right\} P_0(\theta_i^*). \tag{2.20}$$

There are some computational issues that must be highlighted. In the updating step of the non-allocated jumps it is necessary to sample from distribution $\rho_{u,\varepsilon}$ (see (2.19)) which does not belong to any popular family of distributions. In order to sample from $\rho_{\varepsilon,u}(\cdot)$ we used an acception-rejection method. First notice that it is possible to find a couple $(M, g(x))$, with $M > 0$ and $g(x)$ an instrumental distribution, such that $f(x) < Mg(x)$, for every x in $\mathbb{R}$: in particular,

$$M = \frac{e^{-\varepsilon(u+\omega)}}{(\omega+u)^\sigma \Gamma(-\sigma, \varepsilon(\omega+u))\sigma\varepsilon^\sigma}$$

and

$$g(x) = \sigma\varepsilon^\sigma x^{-1-\sigma}\mathbb{1}_{(\varepsilon,+\infty)}(x).$$

Therefore, to the end of sampling a X from $\rho_{u,\varepsilon}$ we have to perform the following steps:

i. Sample Y from $g(y)$

ii. Accept Y as a sample with probability $f(Y)/Mg(Y)$. In other words, sample U from $Unif(0,1)$: if $U \leqslant f(Y)/Mg(Y) = exp(-(u+\omega)(Y-\varepsilon))$ then put X=Y, else return to i.

Note that in order to perform the step i we exploited an inversion of the cumulative distribution function.

Another not straightforward step in the Gibbs Sampler is the updating of ε from the law (2.18). To do so, at each iteration of the algorithm a step of Random Walk Metropolis Hastings is performed. More specifically, given the current value of ε, we sample a new value from the proposal distribution

$N(\varepsilon, 0.5S)$, where S is the support of the prior distribution $\pi(\varepsilon)$. Then the new value is accepted with probability equal to the ratio

$$\alpha = \frac{\pi(\varepsilon^{new}|u, \boldsymbol{\theta}, \mathbf{X})}{\pi(\varepsilon|u, \boldsymbol{\theta}, \mathbf{X})}$$

since we choose a proposal that is symmetrical.

The same technique is been used in the case of random σ: it will be clarified later in this section.

Suppose the kernel density K is gaussian, $K(\cdot|\theta) = N(\cdot|\mu, \sigma^2)$: in this case the latent variable θ is the vector (μ, σ^2). In Chapter 3 we are going to show some numerical results considering this kernel; the most convenient P_0 for the ε-NGG is then

$$P_0(d\mu, d\sigma^2) = P_0^1(d\mu|\sigma^2)P_0^2(d\sigma^2) = N(d\mu|m_0, \sigma^2/k_0)IG(d\sigma^2|a, b),$$

since it is conjugate with the kernel. In fact, the posterior law (2.20) of the i-th allocated point (μ_i^*, σ_i^{2*}) is still a Normal-InvGamma distribution where the parameters are updated as follows:

$$N\left(d\mu_i^*|\left(\frac{m_0 + \sum_{j\in C_i} X_j}{n_i + 1}\right), \frac{\sigma_i^{2*}}{n_i + 1}\right) IG(d\sigma_i^{2*}|a^*, b^*)$$

where

$$a^* = a + \frac{n_i}{2}$$

and

$$b^* = b + \frac{(n_i + 1)}{2}\left[\left(\frac{m_0^2 + \sum_{j\in C_i} X_j}{n_i + 1}\right) - \left(\frac{m_0 + \sum_{j\in C_i} X_j}{n_i + 1}\right)^2\right].$$

Until now we developed the algorithm only in the case of random ε: we want to extend the method when also the parameters σ and κ are random. In both cases the only step to add to the algorithm is the updating of these parameter: the other steps remain the same.

In particular, it is possible to prove that the posterior of the parameter σ is the following:

3′. $\mathcal{L}(\sigma|\varepsilon, \kappa, \mathbf{X}, u, \boldsymbol{\theta}) \propto$

$$\frac{(u + \omega)^{k\sigma}}{\omega^\sigma} \frac{\Lambda_{\varepsilon,u} + k}{\Gamma(-\sigma, \omega\varepsilon)} \prod_{i=1}^{k} \Gamma(n_i - \sigma, \varepsilon(u + \omega)) e^{(\Lambda_{\varepsilon,u} - \Lambda_\varepsilon)} \Gamma(1 - \sigma)^{1-k} \pi(\sigma).$$

.

Clearly, in order to update σ with the previous law we used a Metropolis Hastings algorithm, as for ε.

Moreover, letting κ be a random variable with prior distribution $\pi(\kappa) = Gamma(\alpha, \beta)$, we obtain the following posterior distribution, that is a mixture of gamma densities:

$$\mathbf{3''.} \quad \mathcal{L}(\kappa|\varepsilon, \sigma, \omega, \mathbf{X}, u, \boldsymbol{\theta}) = p_1 G(\alpha + k, R + \beta) + (1 - p_1) G(\alpha + k - 1, R + \beta) \tag{2.21}$$

where G stands for a gamma distribution,

$$R = \frac{\omega^\sigma \Gamma(-\sigma, \varepsilon\omega)}{\Gamma(1-\sigma)} - \frac{(\omega+u)^\sigma \Gamma(-\sigma, \varepsilon(\omega+u))}{\Gamma(1-\sigma)}$$

and

$$p_1 = \frac{(\alpha + k - 1)(u+\omega)^\sigma \Gamma(-\sigma, \varepsilon(\omega+u))}{(\alpha + k - 1)(u+\omega)^\sigma \Gamma(-\sigma, \varepsilon(\omega+u)) + k(R+\beta)\Gamma(1-\sigma)}.$$

Note that the posterior (2.21) is well defined only if $R \geqslant 0$, since $\beta > 0$. Specifically, we need the function $x^\sigma \Gamma(-\sigma, \varepsilon x)$ is decreasing with x. This is true because, omitting the positive constant term $\Gamma(1-\sigma)$,

$$\frac{d}{dx}\left(x^\sigma \int_{\varepsilon x}^{+\infty} t^{-1-\sigma} e^{-t} dt\right) = x^\sigma \left(-(\varepsilon x)^{-1-\sigma} e^{-\varepsilon x}\right) + \sigma x^{\sigma-1} \Gamma(-\sigma, \varepsilon x) =$$
$$= -x^{\sigma-1} \Gamma(1-\sigma, \varepsilon x) \leqslant 0, \ \forall x > 0,$$

where we used the relation $\Gamma(\alpha+1, x) = \alpha\Gamma(\alpha, x) + x^\alpha e^{-x}$.

We open here a parenthesis about the difficulty of the computation of the incomplete gamma when the second parameter is high: this happens, in particular, when the numerosity of the dataset is large because the variable u tends to assume larger values (its mean is n/T_ε), therefore $(u+\omega)$ will be often a big value. The following asymptotic approximation in this case is used:

$$\Gamma(a, x) \simeq e^{-x} x^{a-1} \sum_{m=0} \frac{\Gamma(a)}{\Gamma(a-m)} x^{-m}, \quad x \to +\infty.$$

In conclusion we highlight that thanks to the fact that conditionally on ε only a finite number of components of the mixture has to be considered, namely the support of the mixing measure becomes finite, we do not analytically integrate out the mixing component but impute the $\varepsilon-$NGG prior and update it as a component of the Gibbs sampler thus pursuing a full nonparametric Bayesian inference, obtaining posterior estimates of linear and non

linear functionals of the population distribution. Below, for sake of clarity, a scheme of the Gibbs Sampler which has just been derived is reported.

0. Inizialization

Repeat steps 1-8 for g in 1 ... **G**:

1. Sample $\mathbf{u}^{(g)}$ from a Gamma(n, T_ε)

2. For i=1,..,n sample $\boldsymbol{\theta_i}^{(g)}$ from a discrete distribution s.t. $\mathbb{P}(\theta_i = \tau_j) \propto J_j K(X_i|\tau_j), \quad j = 0, .., N_\varepsilon.$

3. Sample $\boldsymbol{\varepsilon}^{(g)}$ from
$\mathcal{L}(\varepsilon) \propto \prod_{i=1}^{k} \Gamma(n_i - \sigma, \varepsilon(u+\omega)) e^{\Lambda_{\varepsilon u} - \Lambda_\varepsilon} \dfrac{\Lambda_{\varepsilon u} + k}{\Gamma(-\sigma, \omega\varepsilon)} \pi(\varepsilon)$

3'. Sample $\boldsymbol{\sigma}^{(g)}$ from $\mathcal{L}(\sigma) \propto \frac{(u+\omega)^{k\sigma}}{\omega^\sigma} \dfrac{\Lambda_{\varepsilon u} + k}{\Gamma(-\sigma, \omega\varepsilon)} e^{\Lambda_{\varepsilon u} - \Lambda_\varepsilon}$
$\cdot \prod_{i=1}^{k} \Gamma(n_i - \sigma, \varepsilon(u+\omega)) \Gamma(1-\sigma)^{1-k} \pi(\sigma)$

⟵ In case of random parameters

3''. Sample $\boldsymbol{\kappa}^{(g)}$ from a mixture of Gamma densities:
$p_1 \text{Gamma}(\alpha + k, R + \beta) + (1 - p_1)\text{Gamma}(\alpha + k - 1, R + \beta).$

4. Sample $N_{na}^{(g)}$ from
$\dfrac{\Lambda_{\varepsilon u}}{\Lambda_{\varepsilon u} + k} Poi(N_{na} - 1|\Lambda_{\varepsilon u}) + \dfrac{k}{\Lambda_{\varepsilon u} + k} Poi(N_{na}|\Lambda_{\varepsilon u}),$
then set $\boldsymbol{N_\varepsilon}^{(g)} + 1 = N_{na}^{(g)} + k^{(g)}$.

5a. Non-allocated jumps: sample iid from $\mathcal{L}(J_j) \propto e^{-uJ_j} \rho_\varepsilon(J_j)$.

5b. Allocated jumps: sample iid from
$\mathcal{L}(J_{l_i^*}) \propto Gamma(n_i - \sigma, u + \omega) \mathbb{1}_{(\varepsilon, \infty)}.$

6a. Non-allocated points of support: sample iid from P_0.

6b. Allocated points of support: sample iid from
$\mathcal{L}(\tau_i^*) \propto \{\prod_{j \in C_i} K(X_j|\tau_i)\} P_0(\tau_i).$

2.5 | Comparison to Muliere and Tardella's approximation of Dirichlet processes

We describe in this section a similar approach for approximating the Dirichlet prior: here we call this approximation ε_{MT}-Dirichlet distribution, where the subscript MT stands for the initials of the authors of the paper Muliere and Tardella (1998). Based on Sethuraman's stick-breaking representation of a Dirichlet process (Sethuraman, 1994), they introduced a method based on a randomly stopping procedure different from ours.

In particular, the *stick-breaking* representation of the Dirichlet process $DP(\kappa, P_0)$ is the following:

$$P(\cdot) = \sum_{i=1}^{\infty} p_i \delta_{Y_i}(\cdot), \tag{2.22}$$

where $Y_i \overset{iid}{\sim} P_0$ and the weights are built according to the stick-breaking procedure:

$$\begin{aligned} p_1 &= V_1, \\ p_i &= V_i \left(1 - V_{i-1}\right) \dots \left(1 - V_1\right) \quad \forall i \geq 2, \end{aligned}$$

where $V_i \overset{iid}{\sim} Beta(1, \kappa)$, independent on the Y_is. Muliere and Tardella approximate this distribution stopping the series in (2.22) after a random number of terms $n_{\varepsilon_{MT}}$ such that the remaining probability mass is smaller than ε_{MT}.

Definition 4 (ε_{MT}-Dirichlet random probability)**.** *For any $\varepsilon_{MT} \in (0, 1)$, an ε_{MT}-Dirichlet random probability is defined as*

$$P_{\varepsilon_{MT}}(\cdot) = \sum_{i=1}^{n_{\varepsilon_{MT}}} p_i \delta_{Y_i}(\cdot) \cdot \left(1 - r_{n_{\varepsilon_{MT}}}\right)^{-1}$$

where

$$\begin{aligned} n_{\varepsilon_{MT}} &= \inf\{m \in \mathbb{N} : \sum_{i=1}^{m} p_i > 1 - \varepsilon_{MT}\}, \\ r_{\varepsilon_{MT}} &= 1 - \sum_{i=1}^{n_{\varepsilon_{MT}}} p_i. \end{aligned}$$

In this definition the remaining mass $r_{\varepsilon_{MT}}$ is spread proportionally to the sampled part of the distribution. This kind of approximation allows

generating a random probability measure as close as one wants (in total variation distance, as proved in the paper) to the Dirichlet process. Lemma 3 in Muliere and Tardella (1998) states that $n_{\varepsilon_{MT}} - 1$ is Poisson distributed with mean $-\kappa \log(\varepsilon_{MT})$.

This result is very similar to ours which states that $N_\varepsilon \sim Poisson(\Lambda_\varepsilon)$ where $\Lambda_\varepsilon = -\kappa Ei(-\omega\varepsilon)$ (Ei is the exponential integral). However we can not relate formally the two types of approximations because of the construction behind the weights: we have seen that the components p_i of the Muliere and Tardella's approximation are exactly the first $n_{\varepsilon_{MT}}$ weights of the stick-breaking procedure. This implies an "almost decreasing" order, in the sense that they are decreasing in mean:

$$\mathbb{E}(p_{i+1}) < \mathbb{E}(p_i) \quad i = 1, 2, ..$$

For instance, this is not true for our approximation, where the jumps J_i are sampled iid from a Poisson process with intensity $\rho_\varepsilon(x) = x^{-1}e^{-\omega x}(-\frac{1}{Ei(-\omega\varepsilon)})$ in the interval $(\varepsilon, +\infty)$, therefore they are not ordered.

Chapter 3 | Galaxy data

In this chapter we apply the model and the algorithm (implemented in C++ language) proposed in Chapter 2 to a dataset very popular in the literature: the Galaxy dataset. We perform an extensive robustness analysis through a lot of experiments which highlight the relationships between the posterior estimates and the prior choice of the parameters. In fact, the choice of a value (or a prior in the random case) for these parameters remains the most complicated part of the model, since it deeply influences the posterior inference.

3.1 | Description of the robustness analysis

As already pointed out, the dataset used in this chapter is the Galaxy dataset. These data are observed velocities of $n = 82$ different galaxies, belonging to six well-separated conic sections of space. Values are expressed in [Km/s], scaled by a factor of 10^{-3}. The error from sampling the velocities is estimated to be less than 50 Km/s.

We report here the specific model: an ε-NGG mixture model with Gaussian kernel densities. As far as the parameter P_0 of the nonparametric prior concerns, it is fixed as a normal inverse-gamma distribution. Briefly:

$$\begin{cases} X_1, X_2, \ldots, X_n | \theta_1, \theta_2, \ldots, \theta_n \overset{ind}{\sim} N(X_i|\theta_i), \quad \theta_i = (\mu_i, \sigma_i^2) \\ \theta_1, \theta_2, \ldots, \theta_n | P_\varepsilon \overset{iid}{\sim} P_\varepsilon \\ P_\varepsilon \sim \varepsilon - NGG(\kappa, \sigma, \omega, P_0) \\ P_0(d\mu, d\sigma^2) = N(d\mu|m_0, \dfrac{\sigma^2}{k_0}) IG(d\sigma^2|a, b) \\ \varepsilon, \sigma, \kappa \sim \pi_1(\varepsilon)\pi_2(\sigma)\pi_3(\kappa) \end{cases}$$

where $N(X|\mu, \sigma^2)$ represents the univariate normal distribution with mean μ and variance σ^2; $IG(a, b)$ stands for the inverse-gamma distribution with

mean $\frac{b}{a-1}$ (for $a > 1$) and variance $\frac{b^2}{(a-1)^2(a-2)}$ (for $a > 2$). From now on the parameter ω is fixed equal to 1.

Later on we will specify the prior elicited for ε, σ and κ: its effect will be clarified in the next sections.

An extensive set of tests has been done: we report it in Table 3.1. We denote the different "experiments" with the letters A-N and the number after the letters represents the different set of hyperparameters.

For all the tests, we selected as hyperparameters for P_0:

$$\boldsymbol{m_0} = \bar{X} = 20.8315, \quad k_0 = 0.01, \quad a = 2, \quad b = 1$$

which is a standard set of hyperparameters, first proposed by Escobar and West (1995). The value $\boldsymbol{m_0}$ is set equal to the sample mean of the data. We are going to analyze the posterior inference in the next sections.

In Table 3.1, B stands for the Beta distribution and G for the Gamma distribution, while δ is the minimum between 0.1 and the expected value of the sum of the jumps in the NGG process: $\mathbb{E}(T) = \kappa\omega^{\sigma-1}$.

$\boldsymbol{E(K_n)}$	$\boldsymbol{\sigma}$	$\boldsymbol{\kappa}$
3	0.001	0.45
3	0.1	0.25
3	0.2	0.05
5	0.001	1.0
5	0.2	0.35
5	0.3	0.09
20	0.2	5.0
20	0.4	2.2
20	0.6	0.3

Table 3.2: Couples of parameters (σ, κ) fixed for tests D, E, F: we selected three different couples for each prior mean number of groups in the data: $(3, 5, 20)$.

Prior for σ	**Prior for κ**
Beta(2,5)	Gamma(2,2)
Beta(10,23)	Gamma(1.1,8)
Beta(1.1,30)	Gamma(1.1,8)
Beta(1.1,30)	Gamma(100,50)
Beta(10,23)	Gamma(100,50)

Table 3.3: Couples of priors for parameters (σ, κ) in tests M: we selected different prior information for the parameters and, consequently, for the number of clusters.

In order to correctly evaluate the model for density estimation proposed in this work, one could be interested in the time employed by the algorithm to run the code. In Tables 3.4, 3.5, 3.6, 3.7 there are the run-times of each test made on a processor Intel Core i7 2670QM with 6GB of RAM. Every run produces a final sample of 10000 iterations, after a thinning of 10 and an initial burn-in period of 10000 iterations.

Name	ε	σ	κ
A0, ..., A8	10^{-6}	$\{0.001, 0.1, \ldots, 0.8\}$	0.45
B0, ..., B9	$Unif\,(0, \delta)$	$\{0.001, 0.1, \ldots, 0.9\}$	0.45
C0, ..., C9	$Beta\,(0.69, 2.06)$ in $(0, \delta)$	$\{0.001, 0.1, \ldots, 0.9\}$	0.45
D0, ..., D8	10^{-6}	As in Table 3.2	As in Table 3.2
E0, ..., E8	$Unif\,(0, \delta)$	As in Table 3.2	As in Table 3.2
F0, ..., F8	$Beta\,(0.69, 2.06)$ in $(0, \delta)$	As in Table 3.2	As in Table 3.2
G0, ..., G6	10^{-4}	$B\,(2, 18)$	$\{0.01, 0.05, 0.07, 0.1, 0.5, 1, 2\}$
H0, ..., H3	10^{-4}	$B(1, 19)$,$B(1.5, 13.5)$,$B(3, 7)$,$B(2, 2)$	0.45
I0, ..., I9	10^{-4}	$\{0.001, 0.1, \ldots, 0.8, 0.9\}$	$G(2, 2)$
L0, ..., L3	10^{-4}	0.001	$G(1.1, 2)$,$G(2, 2)$,$G(5, 3)$,$G(10, 3)$
M0, ..., M5	10^{-4}	As in Table 3.3	As in Table 3.3
N0, ..., N3	$\{10^{-6}, 10^{-3}, 10^{-1}, 1\}$	0.4	0.45

Table 3.1: Scheme of the tests in the robustness analysis.

It is obvious that the value of ε greatly influences the run-time of the algorithm: see, for instance, tests N, where the variable ε is considered fixed and increases from 10^{-6} to 1 (Table 3.7). The time greatly decreases from N0 to N3: it goes from almost 8 minutes to 21 seconds.

Test name	Time	Test name	Time	Test name	Time
A0	1m28s	B0	48s	C0	1m1s
A1	56s	B1	31s	C1	36s
A2	1m30s	B2	35s	C2	39s
A3	2m53s	B3	38s	C3	43s
A4	7m01s	B4	43s	C4	51s
A5	17m13s	B5	48s	C5	1m3s
A6	39m46s	B6	50s	C6	1m2s
A7	120m50s	B7	55s	C7	1m7s
A8	280m26s	B8	56s	C8	1m8s
		B9	59s	C9	1m7s

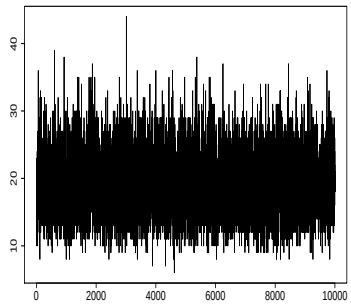
Table 3.4: Run-times of the algorithm in tests A, B, C: in group A ε is fixed, in B a Uniform prior on $(0, \delta)$ for ε is assumed while in C the prior is a Beta distribution.

Moreover, comparing tests A, B and C, we see that it is computationally convenient to consider a random ε rather than a fixed and small one, as in tests of group A (Table 3.4): the times in the first column are higher than those in the second and third columns, even though in B and C there is an additional step that updates the variable ε to be done. It is possible to appreciate this fact also in tests of group D, E, F in Table 3.5. In particular, in group E the variable ε is free to assume also relatively large values since the prior is uniform: comparing these times with the same tests of group D, where ε is fixed to a very small value, it is perceptible a substantial gain in computational time. We will realize in next sections that this occurs because a small value of ε implies a lot of components to consider in the mixture.

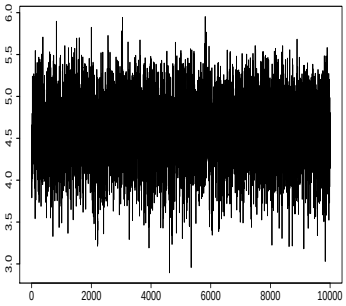
On the other hand, large values for σ imply more elements to consider in the mixture. Hence, the computational time increases: look, for example, tests A, B, C in Table 3.4 where σ is considered on a grid from 0.001 to 0.99 and κ is fixed. The same happens in tests of group I in Table 3.6.

From Table 3.5 we see that assuming a smaller $\mathbb{E}(K_n)$ implies a gain in run-time, since of course the algorithm must take into account less clusters.

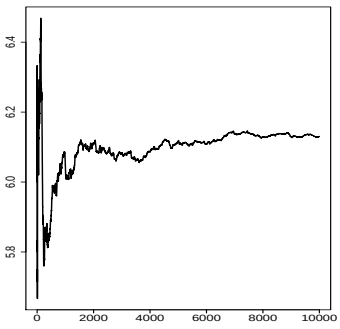
Convergence has been checked for every run monitoring the traceplots of posterior chains and the cumulate mean number of clusters: we report an example for test A8 and A0 in Figure 3.1. Figures 3.1 (a) and (b) show satisfactory posterior chains of variables K_n and U, Figure 3.1 (c) illustrates



(a) Traceplot of the number of groups in test A8.



(b) Traceplot of the latent variable U in test A8.



(c) Iteration versus cumulate mean number of groups in test A0.

Figure 3.1: Some convergence indexes.

Test name	Time	Test name	Time	Test name	Time
D0	1m28s	E0	1m10s	F0	1m1s
D1	42s	E1	43s	F1	35s
D2	34s	E2	36s	F2	30s
D3	2m41s	E3	1m35s	F3	1m45s
D4	1m17s	E4	37s	F4	41s
D5	57s	E5	35s	F5	44s
D6	10m45s	E6	1m9s	F6	1m28s
D7	35m15s	E7	59s	F7	1m16s
D8	35m15s	E8	57s	F8	1m20s

Table 3.5: Run-times of the algorithm in tests D, E, F: for the parameters used in these tests see Table 3.2.

Test name	Time	Test name	Time	Test name	Time
G0	28s	I0	4m2s	L0	3m20s
G1	34s	I1	1m18s	L1	4m3s
G2	35s	I2	1m37s	L2	4m52s
G3	39s	I3	2m12s	L3	7m2s
G4	52s	I4	2m50s	M0	2m11s
G5	1m9s	I5	4m10s	M1	1m
H0	59s	I6	5m55s	M2	1m47s
H1	1m	I7	7m30s	M3	1m15s
H2	1m32s	I8	8m15s	M4	4m30s
H3	2m20s	I9	9m15s	M5	45m40s

Table 3.6: Running times of the algorithm in tests G, H, I, L, M.

the cumulate mean number of groups, which is very stable after the burn-in period (the x-axis represents iterations' number).

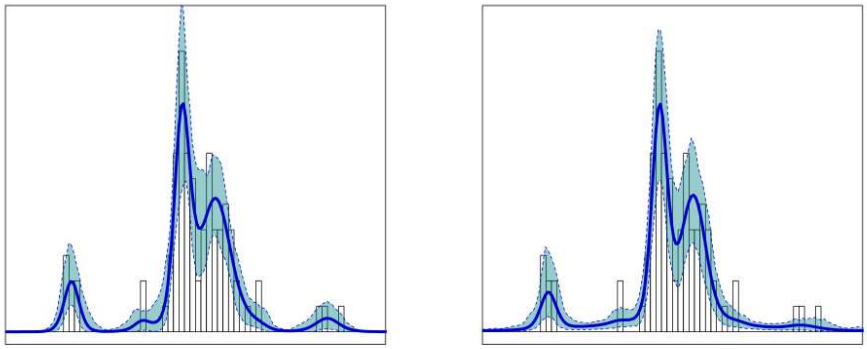
Finally, we underline that all the density estimates are pretty good: we will not present them for each prior we have considered since they are very similar. However, this fact reflects the robustness of the method.

In order to give an estimation of the quantile functional of the density, we refer to Gelfand and Kottas (2002). In their paper, they provide a computational approach about the estimation of a generic functional H for a mixture model. We report the main passages.

Given a mixture distribution of the form $F(\cdot, G) = \int K(\cdot, \theta)G(d\theta)$, the goal is to estimate $H(F(\cdot, G))|data$, where H is a generic functional of F. The simulation provides a posterior chain of the latent variable $\theta_b,\ b = 1, .., B$.

Test name	Time
N0	7m56s
N1	52s
N2	29s
N3	21s

Table 3.7: Running times of the algorithm in tests N.



(a) Density estimation and its pointwise 90% credibility interval for test A0.

(b) Density estimation and its pointwise 90% credibility interval for test A8.

Figure 3.2: Two examples of density estimates with the corresponding quantiles.

If H is linear, then a simple Monte Carlo integration is needed, since

$$H(F(\cdot, G)) = \int H(K(\cdot, \theta)G(d\theta)) \simeq B^{-1} \sum_{b=1}^{B} H(K(\cdot, \theta_b)).$$

On the other hand, if H is non linear we can evaluate G_b, which is a realization from $G|data$: thanks to the Monte Carlo integration we obtain $H_b = H(F(\cdot, G_b)),\ b = 1, .., B$, realization from $H(F(\cdot, G))|data$, from which the estimation for H can be recovered.

In our specific case, where H is the quantile functional ν_p, we evaluate

$$F_b = F(grid, G_b) = \sum_{i=0}^{N_\varepsilon^{(b)}} p_i^{(b)} K(grid, \theta_i^{(b)})$$

for every iteration of the algorithm, over a grid of values where we want to evaluate the functional.

Then, for every value x of the grid, we apply the quantile function ν_p on the B values, obtaining an estimation of the 5% and 95% quantiles.

In Figure 3.2 an example of density estimation is provided: the blue part represents the 90% credible interval (pointwise) computed with the previous method and the thick line is the mean, i.e. the predictive distribution.

3.2 | The ε-NGG mixture model with fixed parameters

First of all we provide an example of density estimation for tests of group N where the parameters are kept fixed and ε increases (Figure 3.3): all the estimates are pretty good and detect the "right" number of clusters, even if the estimates adjust to the data, so there are more groups, when ε is small.

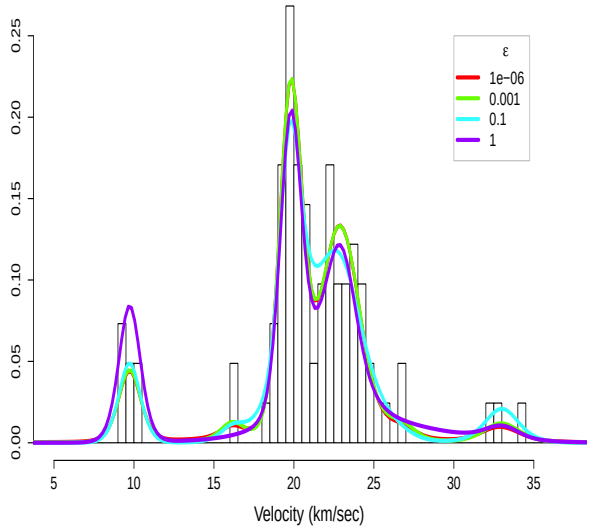
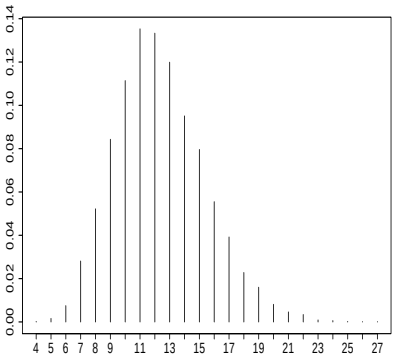
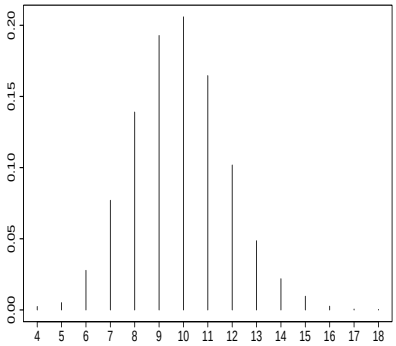


Figure 3.3: Density estimation in tests N with different values for parameter ε.

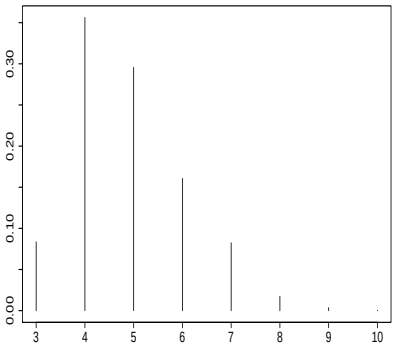
Thanks to the tests in group N we understand the influence of the parameter ε: when ε increases, more jumps are cut off from the sum defining the process and, consequently, less components in the mixture must be considered. Recalling that $N_\varepsilon+1$ is the sum of the number of allocated jumps (the groups) and the non-allocated jumps, it is clear that less clusters are used to describe the data (see Figure 3.4).



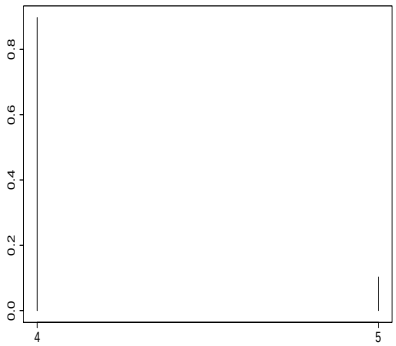
(a) N0: $\varepsilon = 10^{-6}$



(b) N1: $\varepsilon = 10^{-3}$



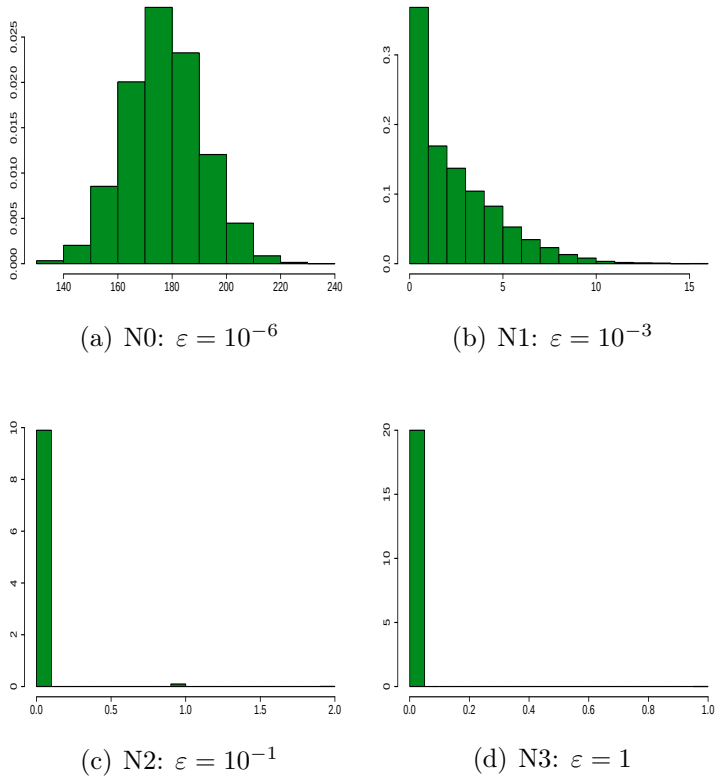
(c) N2: $\varepsilon = 10^{-1}$



(d) N3: $\varepsilon = 1$

Figure 3.4: Histograms of K_n for each test in group N.

A decrease of number of non-allocated jumps is visible from Figure 3.5, when ε increases: thanks to this fact, a huge gain in run-time is reached, as we can see in Table 3.7. In fact, the time goes from approximately 8 minutes when ε is very small to less than 1 minute when the parameter assumes the largest value.



(a) N0: $\varepsilon = 10^{-6}$ (b) N1: $\varepsilon = 10^{-3}$

(c) N2: $\varepsilon = 10^{-1}$ (d) N3: $\varepsilon = 1$

Figure 3.5: Histograms of variable number of non allocated jumps N_{na} for each test of group N.

Another relationship one could be interested in, is that between σ and the posterior number of clusters: as expected, looking at Figure 3.6 related to experiments A, we can observe an approximately linear increasing of the posterior mean number of clusters while σ goes from 0 to the maximum. Of course, also the variance increases.

We also point out that increasing the value of σ has a positive effect on the MCMC chains: see Figure 3.7 to notice the improvement on the autocorrelation of variable U between the test with σ equal to 0.001 and σ equal to 0.8.

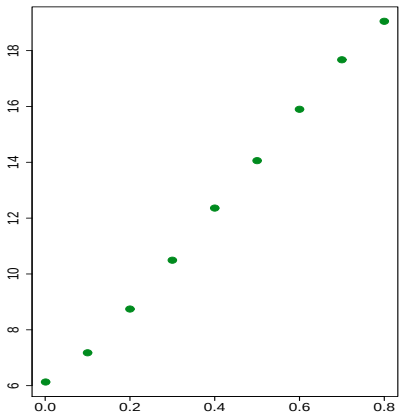


Figure 3.6: Values of the parameter σ versus posterior mean number of clusters K_n in tests A.

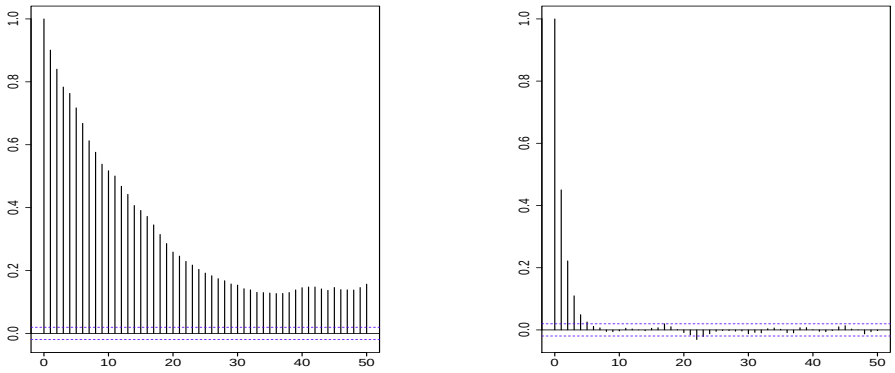


Figure 3.7: Autocorrelation of the auxiliary variable U in tests A0 where $\sigma = 0.001$ (left) and A8 where $\sigma = 0.8$ (right).

3.3 | The ε-NGG mixture model with random parameters

In this section the effect of the randomness of the parameters is studied, in order to choose a suitable prior for the parameters of the process. We divide the section into four subsections: the first one deals with the precision parameter ε, while the second and the third ones are related to the randomness of the parameters of the NGG process σ and κ, respectively. They will be considered both random in Section 3.3.4.

3.3.1 | The effect of the prior on ε

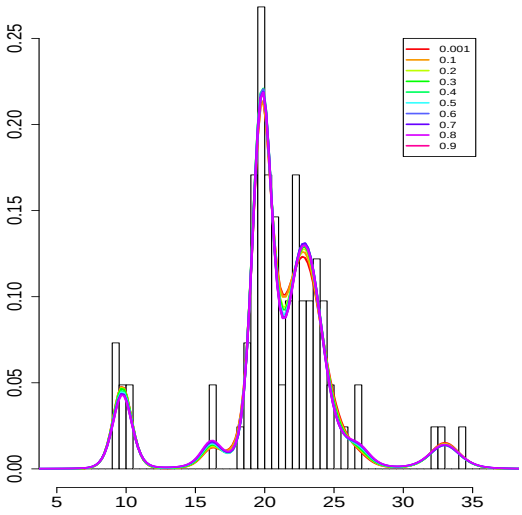


Figure 3.8: Density estimation in tests B with different values of parameter σ.

When ε is random, the model is expected to be more flexible, since it would "adjusts" for the number of jumps of the process P_ε that must be considered. If ε increases the process will be significantly different from the NGG process, since in this case many small jumps will not be included in the mixture. Moreover, when ε is large, the variable N_ε, which counts the number of weights in the mixture, will be generally smaller than when ε is fixed.

The density estimations are good, as we can appreciate in Figure 3.8, but, as we expect, the model is more parsimonious with respect to the variable

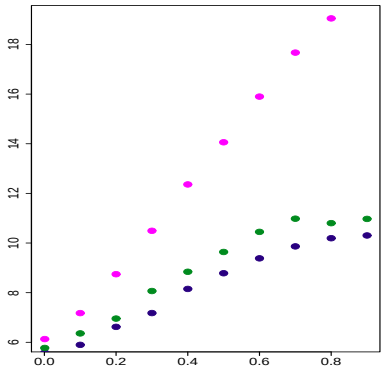


Figure 3.9: Value of the parameter σ versus mean number of clusters in tests A (magenta), B (blue) and C (green).

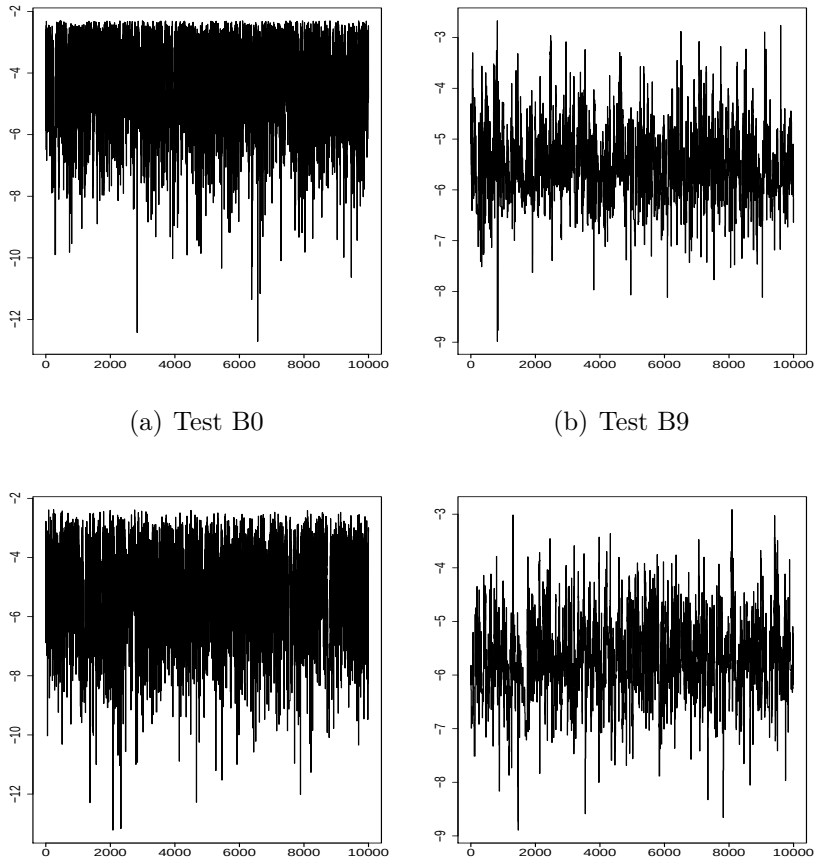
K_n, number of groups, especially in the case with the uniform prior which is the most non informative. This fact can be understood from Figure 3.9 where the posterior mean of K_n diminishes letting ε be random (points green and blue). In particular, we can see that a sort of asymptote in the mean number of cluster is been reached in the random case, because we do not force the model to have a large number of clusters.

It is important to notice that also the number of non-allocated jumps is smaller than the case with fixed ε. This fact, together with the reduction of K_n, explains the huge profit in run-time, Table 3.4. The algorithm is faster since it needs to sample less jumps and less points of support, which is the most complex part from a computational point of view.

Moreover, we notice that the traceplot of the non-allocated jumps gets better increasing σ. On the other hand, observing the variable T_ε that represents the total mass of the process, we have discerned that the chain improves in the random case with respect to the fixed case. The behavior of this variable is significantly different in the two cases, but this fact is not simply explicable.

Examine now the traceplot of $log(\varepsilon)$ in the experiments B e C (Figure 3.10): the chain is better when σ is small, as in cases B0 and C0. As far as the robustness with respect to σ is concerned, we should acknowledge that, as σ increases, more computational problems come up, because of the incomplete gamma function, that is harder to be numerically evaluated. Also the autocorrelation gets worse with large values of σ.

An interesting information we can extract from the MCMC chain of ε is its posterior distribution: looking at Figure 3.11 we deduce the model suggests to assume small values for ε even if the prior is non-informative,



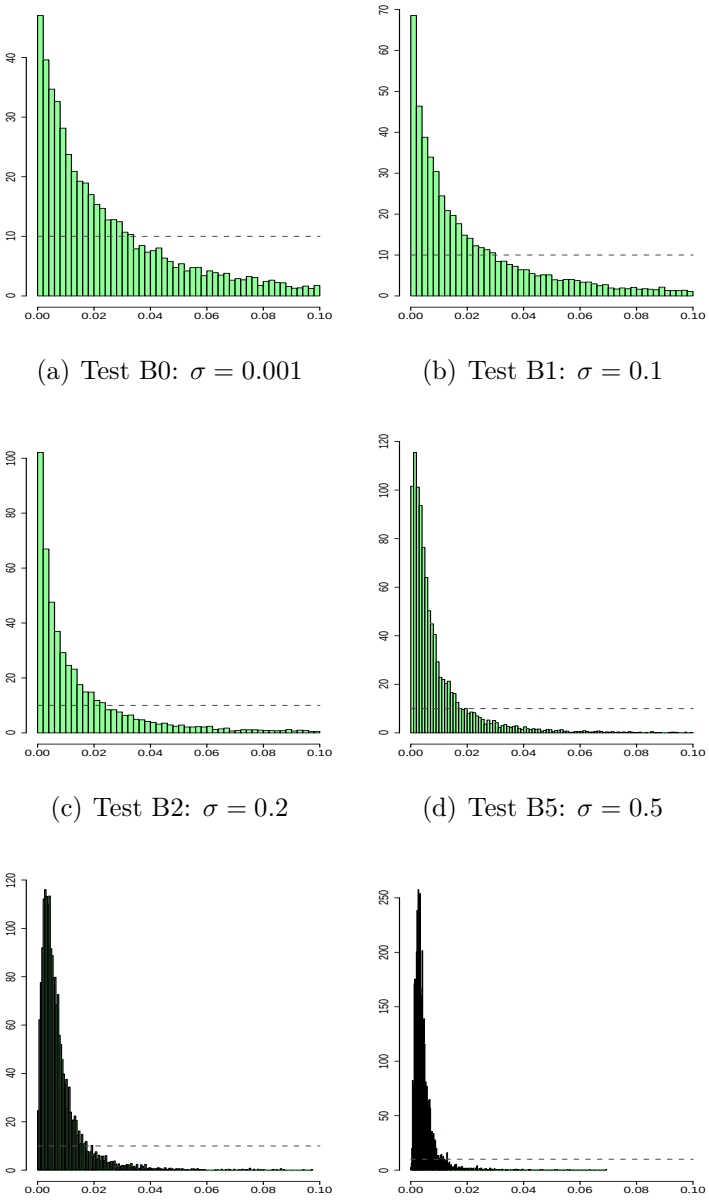
(a) Test B0

(b) Test B9

(c) Test C0

(d) Test C9

Figure 3.10: Traceplot of the variable $\log(\varepsilon)$ in different tests.



(a) Test B0: $\sigma = 0.001$ (b) Test B1: $\sigma = 0.1$

(c) Test B2: $\sigma = 0.2$ (d) Test B5: $\sigma = 0.5$

(e) Test B7: $\sigma = 0.7$ (f) Test B9: $\sigma = 0.9$

Figure 3.11: Histograms of variable ε in different tests of group B with superimposed in gray the prior, $Unif(0, \delta)$.

uniform between 0 and 0.1.

In particular, increasing σ and consequently the expected value of prior number of clusters, smaller values of ε are needed: even if we do not know the exact relation between ε and the number of clusters in the data we deduce from the numerical results that reducing ε the number of groups increases. This fact is simple to imagine since decreasing ε leads to consider a bigger amount of points of support. These considerations hold also for tests of group C, where we gave a prior concentrated over very small values.

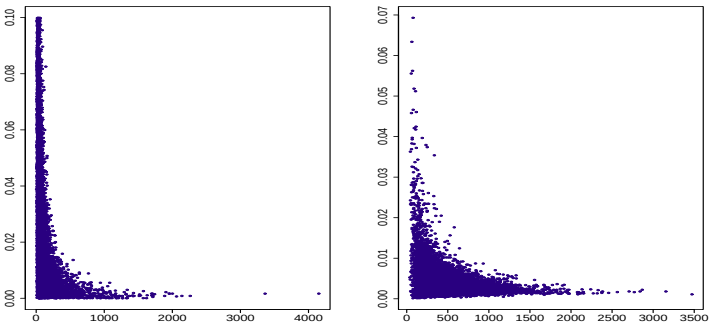


Figure 3.12: Variable U versus ε in test B0 (left) and B9 (right).

We also point out that there exists a strong correlation between the variables U and ε, above all when σ is small: see the scatterplots in Figure 3.12.

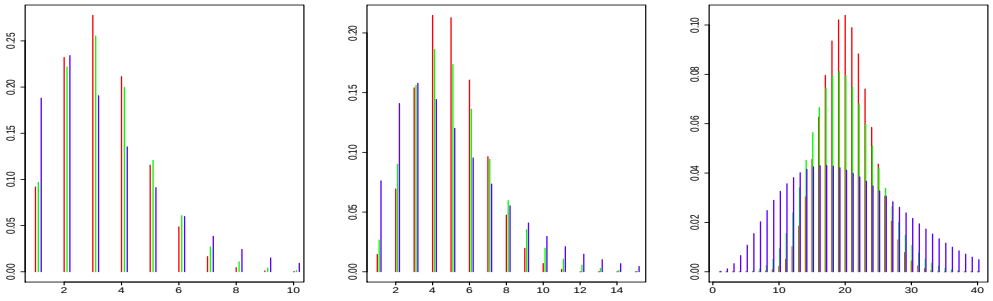


Figure 3.13: Prior distributions of the variable number of clusters: (left) the mean is 3 for all the three different couple of (σ, κ) of Table 3.2 , while the mean is 5 (center) and 20 (right).

In the experiments D, E and F we fixed a priori the mean number of groups to be 3, 5 or 20 (of course in the corresponding NGG case, where

$\varepsilon = 0$, because the distribution is available only in this case). For each expected value we selected three couples (σ, κ) which correspond to the prior distributions for the variable K_n in Figure 3.13: as pointed out in Section 1.5.1, an increase of σ means a prior distribution with a larger variance, so more non-informative.

Consider now the case with ε fixed (test D): we cannot appreciate a significant difference between posterior distributions of K_n in the 3 cases when $\mathbb{E}(K_n) = 3$, Figure 3.14 (a), while it is possible to see it in the other two cases, compare Figures 3.14 (d) and (g).

When σ assumes larger values also posterior distributions of K_n become spread to a larger range of possible values: since the model is more flexible the posterior mean is free to shift towards the "real" average being more "sensible" to the data. In this case the model seems to tell us that there are about 10 clusters, in fact:

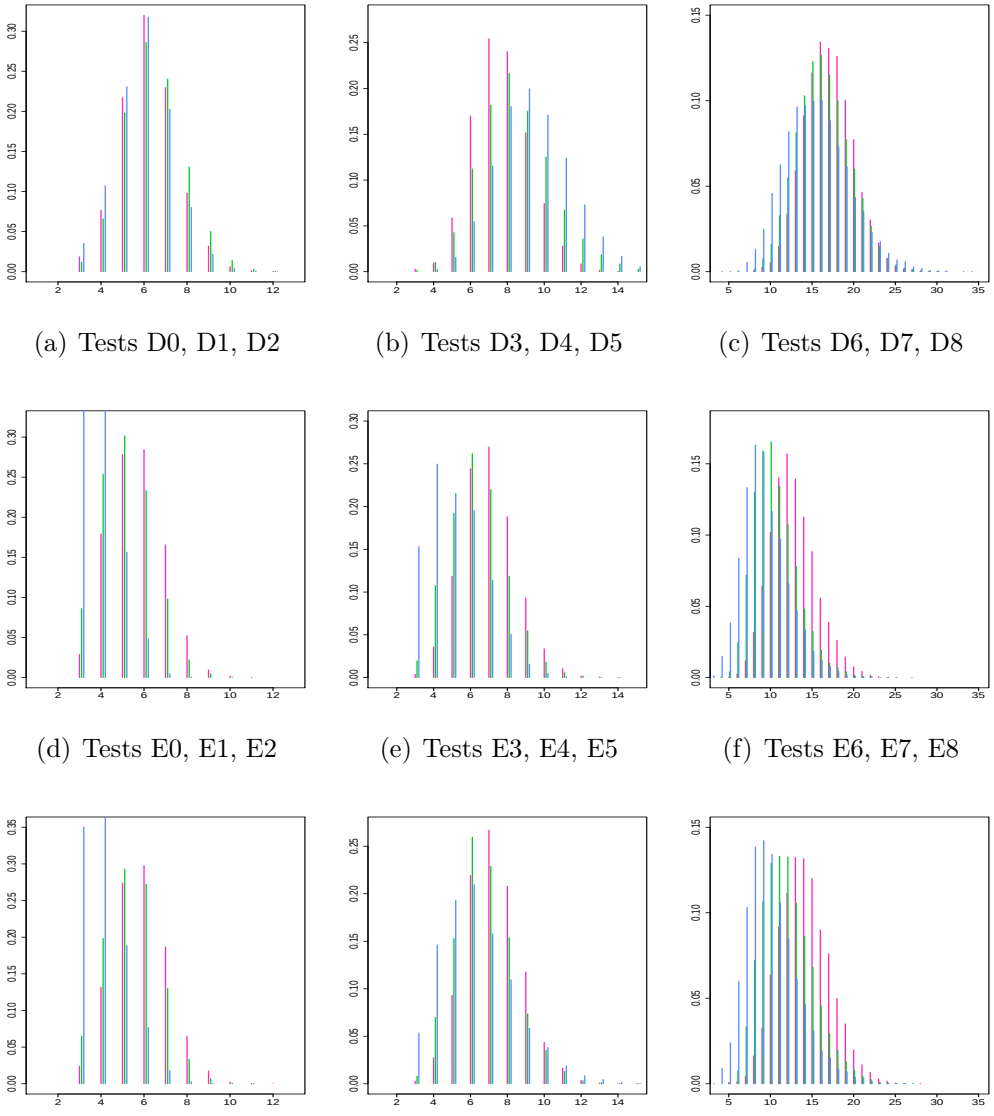
- If $\mathbb{E}(K_n) = 3$, the a-posteriori mode is 6 in all the cases;
- If $\mathbb{E}(K_n) = 5$, the a-posteriori mode is 7,8,9;
- If $\mathbb{E}(K_n) = 20$, the a-posteriori mode is 15, 17, 19.

We notice from the previous values that in all the experiments there is a shift of $\mathbb{E}(K_n|data)$ towards larger values in the case with $\mathbb{E}(K_n) = 3$ or 5, towards smaller values in the case with 20 as prior mean value. This shift is more visible when σ is large, thanks to the flexibility of the model. Moreover, fixed the value for $\mathbb{E}(K_n)$, the number of non-allocated jumps decreases when σ enlarges (and κ diminishes, of course): see Figure 3.15 and observe how the number of non-allocated jumps decreases when the a-priori variance of K_n increases with σ.

Considering now tests E e F with ε random, the model seems to find a lower number of clusters: in the tests with very flexible models the shift of the distribution occurs towards smaller values with respect to the fixed ε case. See for example the histograms (d) to (i) in Figure 3.14. In this case the posterior distributions present a variance relatively small even if the prior variance is very big: the variability affects ε.

It is interesting to notice from the histograms of ε in Figures 3.16 and 3.17 that not always the values of ε are small: in some cases (for example E6 and F6) the model tends to become "parametric", because only few components of the mixture are considered (N_ε tends to K_n, the number of groups). This fact highlights the flexibility of the process which is suitable to represent a large variety of situations.

Moreover, we point out that once $\mathbb{E}(K_n)$ is fixed, increasing σ yields smaller values for ε: see, for instance, Figures 3.17 (g), (h), (i) or 3.16 (d),

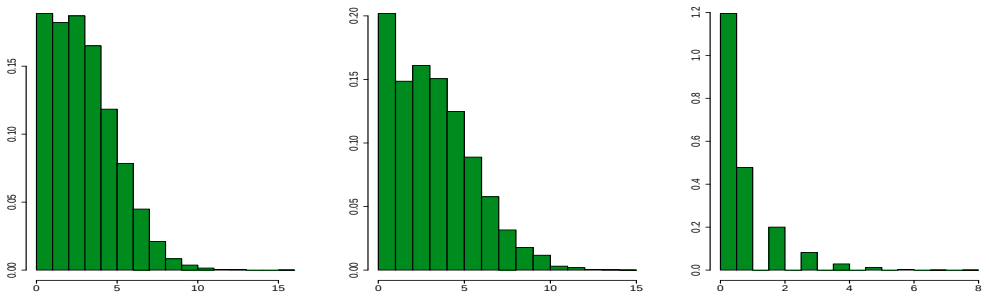


(a) Tests D0, D1, D2 (b) Tests D3, D4, D5 (c) Tests D6, D7, D8

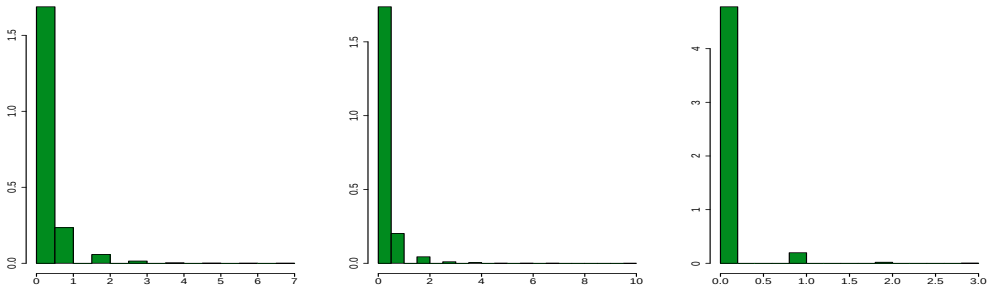
(d) Tests E0, E1, E2 (e) Tests E3, E4, E5 (f) Tests E6, E7, E8

(g) Tests F0, F1, F2 (h) Tests F3, F4, F5 (i) Tests F6, F7, F8

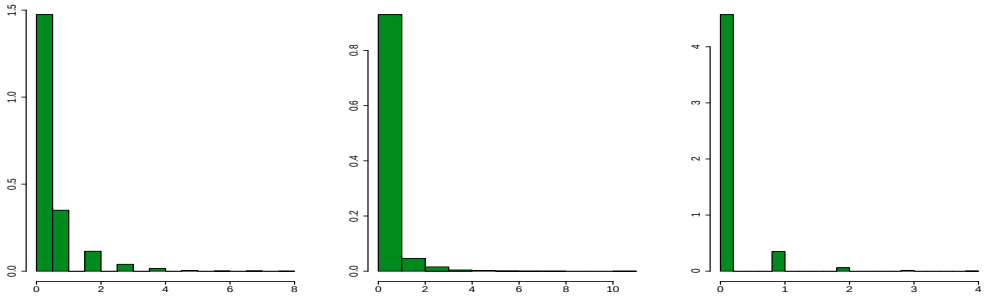
Figure 3.14: Histograms of the posterior number of clusters in tests D, E, F. In blue the tests with a bigger a-priori variance for K_n, in magenta the tests corresponding to a relatively small variance a-priori, in green the intermediate ones.



(a) Tests D0 (b) Tests D1 (c) Tests D2

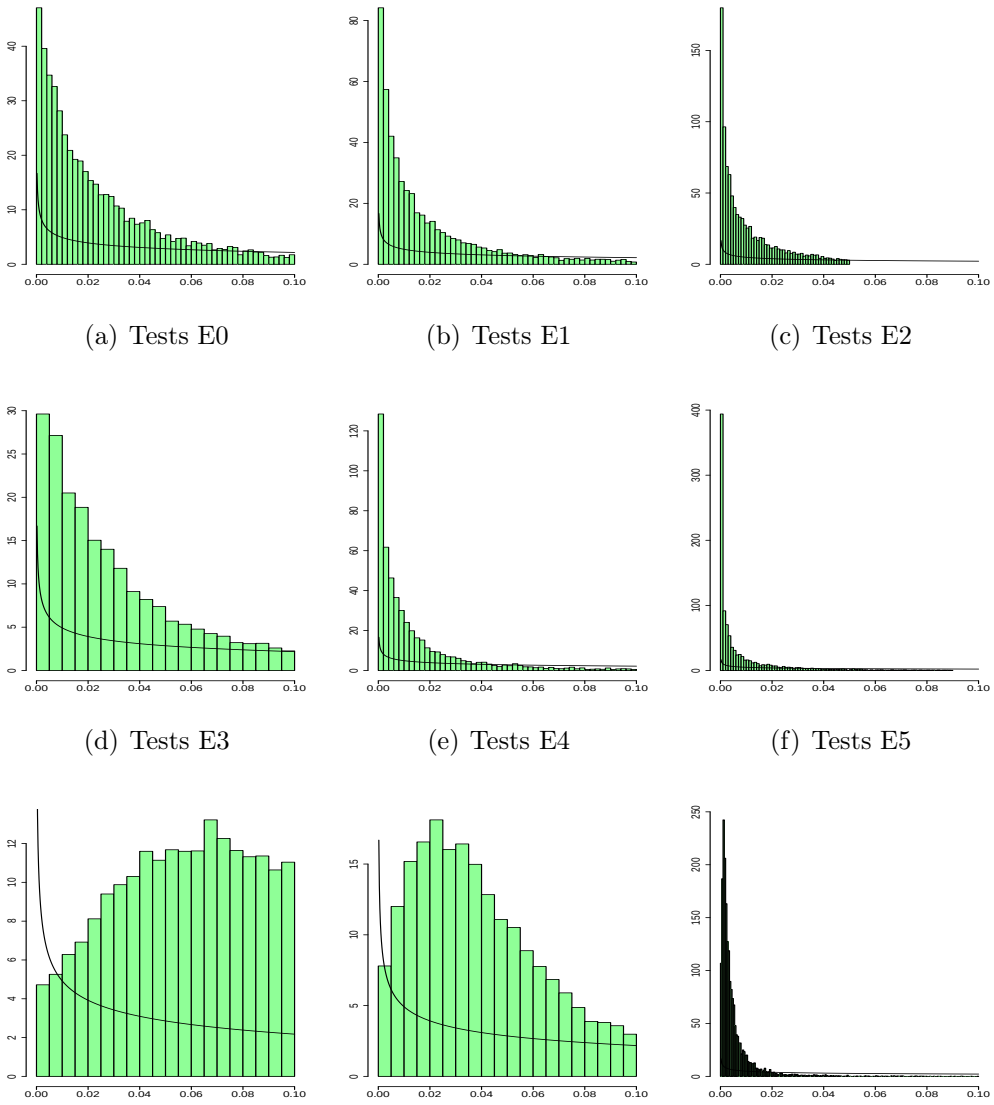


(d) Tests E0 (e) Tests E1 (f) Tests E2



(g) Tests F0 (h) Tests F1 (i) Tests F2

Figure 3.15: Histograms of the variable N_{na}, number of non-allocated jumps, in tests D, E, F where the a-priori mean number of clusters is 3.

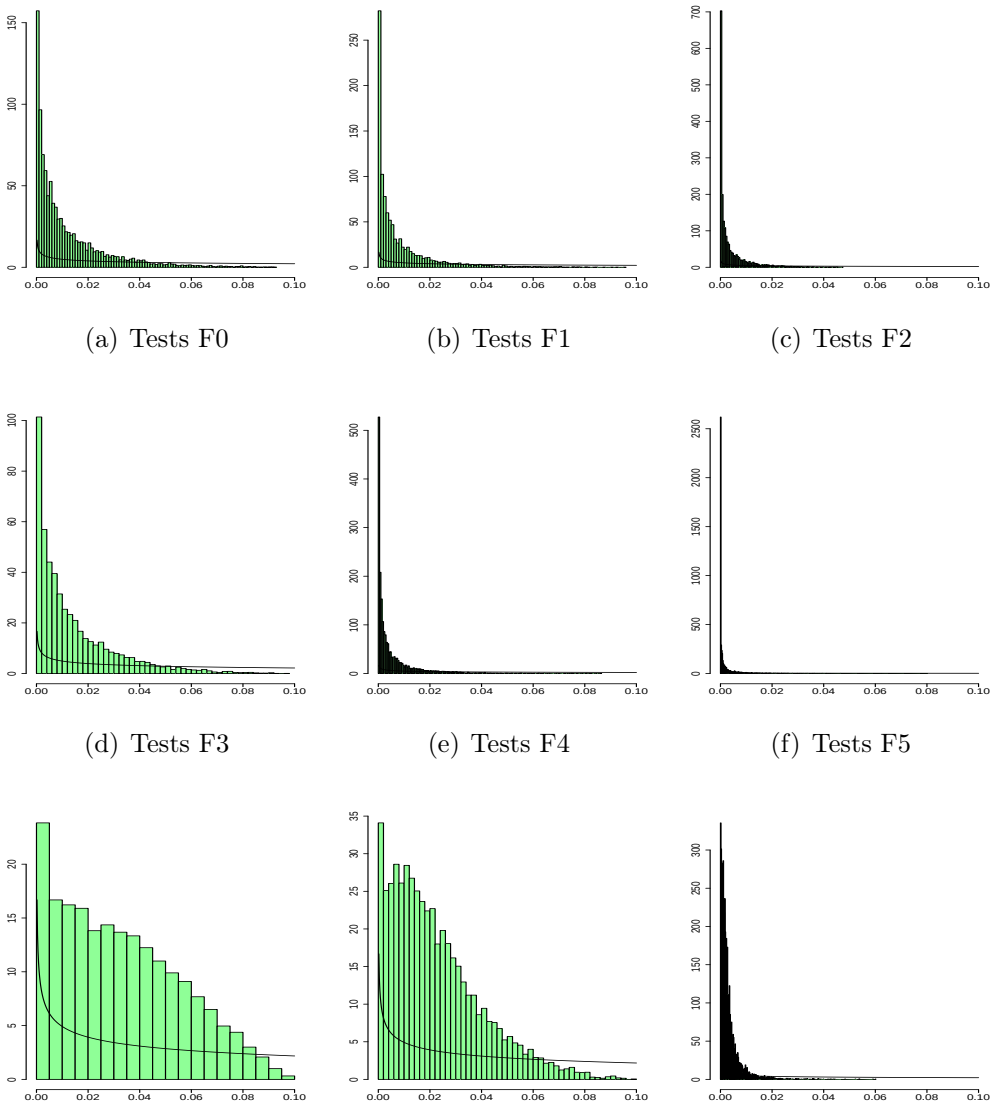


(a) Tests E0 (b) Tests E1 (c) Tests E2

(d) Tests E3 (e) Tests E4 (f) Tests E5

(g) Tests E6 (h) Tests E7 (i) Tests E8

Figure 3.16: Histograms of the variable ε in tests E: in black, the prior.



(a) Tests F0 (b) Tests F1 (c) Tests F2

(d) Tests F3 (e) Tests F4 (f) Tests F5

(g) Tests F6 (h) Tests F7 (i) Tests F8

Figure 3.17: Histograms of the variable ε in tests F: in black, the prior.

(e), (f). This happens because a large value for σ implies more clusters and more non-allocated jumps in the data: a small value for ε is thus needed in order to have enough components in the mixture.

Finally, an interesting issue related to the randomness of ε is that the number of non-allocated jumps diminishes with respect to the non random case since the algorithm is free to consider a larger ε, therefore taking into account less non-allocated jumps (see Figure 3.15).

3.3.2 | The effect of the prior on σ

In this section we analyze the experiments made with random σ, in order to understand better how the choice of the prior affects the estimates. We have already seen that both σ and κ have the effect of increasing the number of clusters in the model. Look for example Figure 3.18 or 3.9.

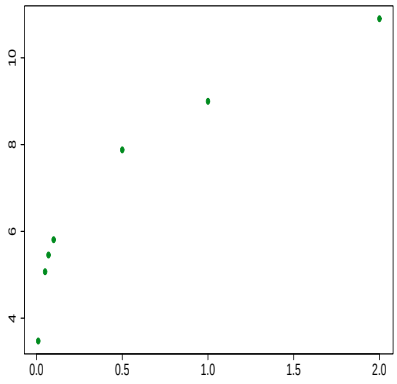


Figure 3.18: Tests of group G: values of variable κ versus the posterior mean number of clusters.

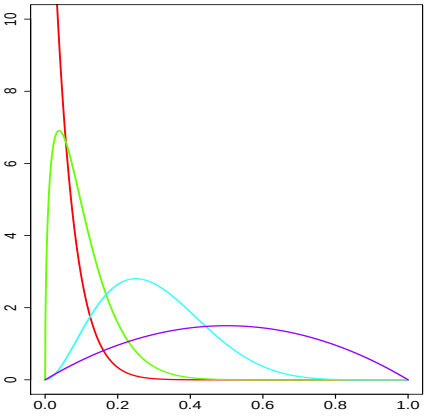
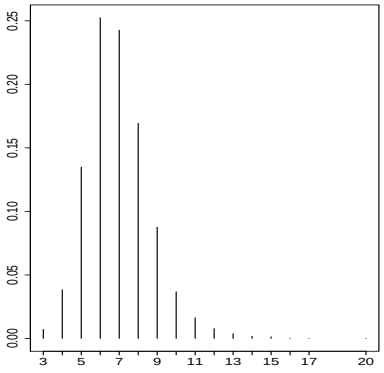


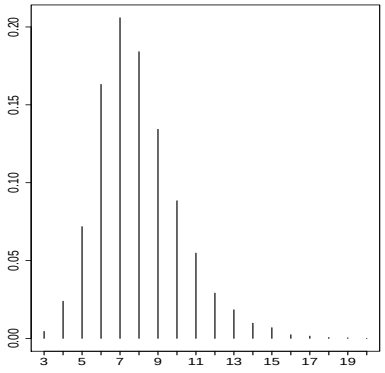
Figure 3.19: Priors for the parameter σ in tests H: Beta(1, 19) (red), Beta(1.5, 13.5) (green), Beta(3, 7) (light blue), Beta(2, 2) (purple).

But what happens letting σ be a random variable? In Figure 3.19 four different priors used for the experiments H are shown: they become more non informative but also shifted towards larger values. As expected the number of clusters rises (Figure 3.20) together with the number of non-allocated jumps which is huge: the computation time explodes.

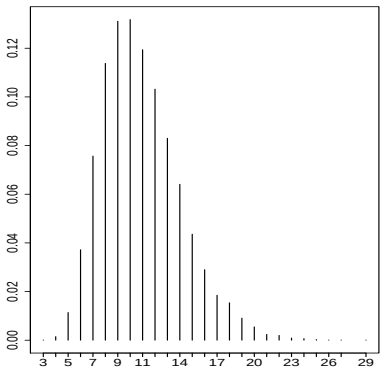
Observing the histograms of σ in Figure 3.21, if the prior is concentrated on small values, as in H0 and H1, the chain tends to move towards larger values ((a) and (b)) while in the case of non informative priors (see (d)) the



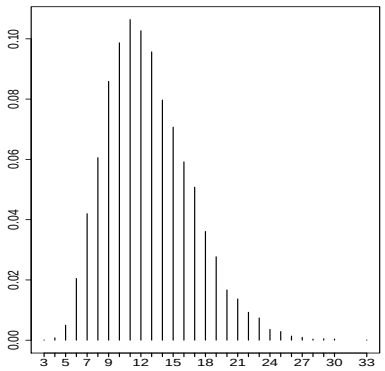
(a) Test H0



(b) Test H1



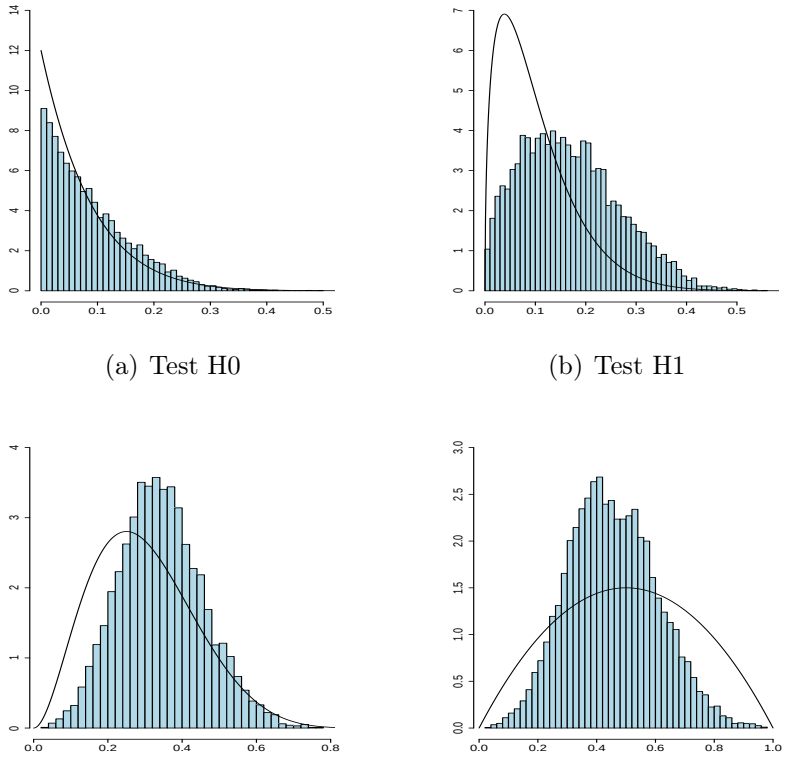
(c) Test H2



(d) Test H3

Figure 3.20: Histograms of K_n in tests of group H.

values of the chain tends to assume intermediate values in the range $(0.2, 0.7)$. In any case, the model tends to be very different from the Dirichlet Process, which corresponds to $\sigma = 0$.

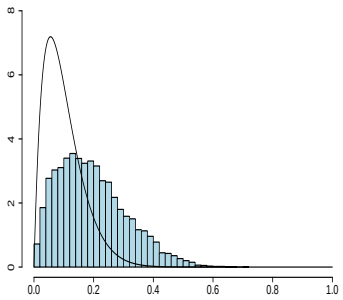


(a) Test H0 (b) Test H1

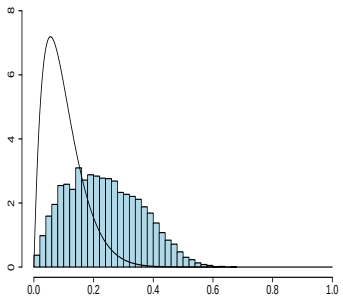
(c) Test H2 (d) Test H3

Figure 3.21: Histogram of σ in tests of group H.

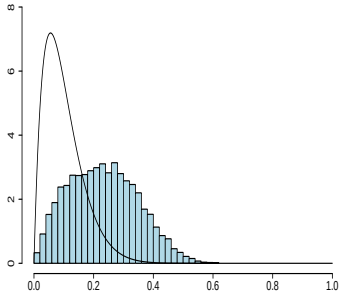
In the experiments G we selected κ on a grid and we assumed a beta distribution as prior for σ: we are going to study the behavior of the variable σ in these conditions. Obviously, the number of groups increases as we have already noticed from Figure 3.18. Observe now Figure 3.22, where posterior distributions of σ are shown: one one hand, when κ is very small (as in cases (a), (b), (c), (d)) σ increases, playing a key role in adjusting the number of clusters and correcting a model that could be too poor. On the other hand, when κ is large enough (and the model is sufficiently rich) σ remains small, as suggested by the prior.



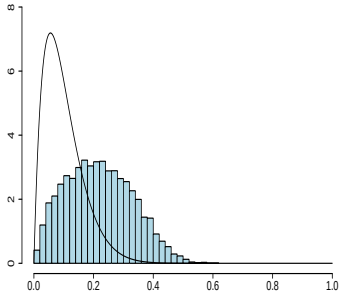
(a) Test G0: $\kappa = 0.01$



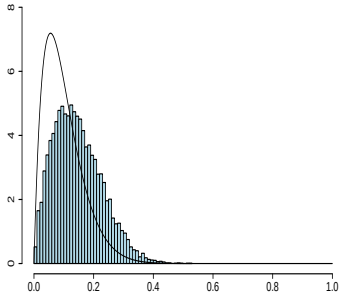
(b) Test G1: $\kappa = 0.05$



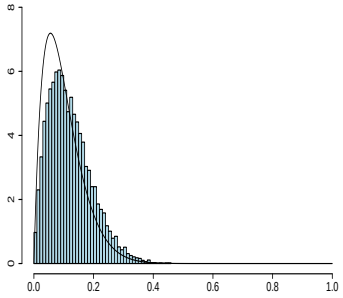
(c) Test G2: $\kappa = 0.07$



(d) Test G3: $\kappa = 0.1$



(e) Test G4: $\kappa = 0.5$



(f) Test G5: $\kappa = 1$

Figure 3.22: Histogram of σ in tests of group G.

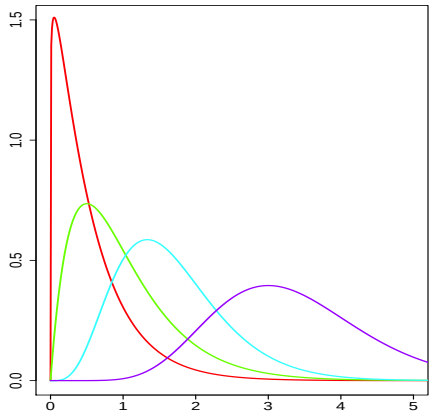


Figure 3.23: Four different priors for the parameter κ in tests of group L: $Gamma(1.1, 2)$ (red), $Gamma(2, 2)$ (green), $Gamma(5, 3)$ (cyan), $Gamma(10, 3)$ (purple).

3.3.3 | The effect of the prior on κ

In this section the influence of the mass parameter κ is studied.

In the set of tests L we have chosen four different priors for the parameter, as in Figure 3.23 .

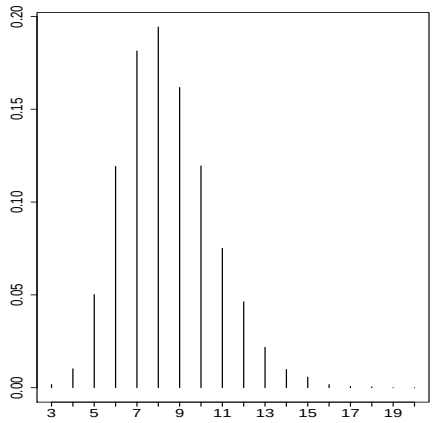
Examining the histograms of the number of clusters K_n, we notice that the randomness of κ has a positive effect on the estimates. Indeed, as we expected, there is an increase of the posterior mean value because of the prior information we gave: therefore, this augment is smoothed thanks to the randomness of κ, since the data can influence more the results (see Figure 3.24).

Studying the histograms of the non-allocated jumps N_{na}, it is clear that they increase, even though not strongly: the parameter κ has a softer effect with respect to σ.

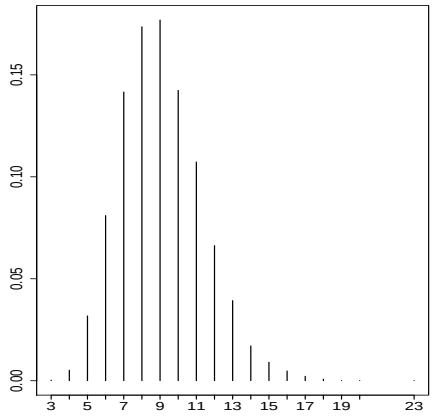
Exactly as the case where σ was random, now κ has the key role of "compensating" the small value of σ, 0.001. The posterior histograms of κ in Figure 3.25 highlight the shift of the posterior distribution towards larger values with respect to the prior distribution in the cases L0, L1 and L2.

With the ε-NGG process we have two parameters compared to the one of the DPM: this is useful since one can compensate the other if one of them is shifted over values that are not suitable for the data.

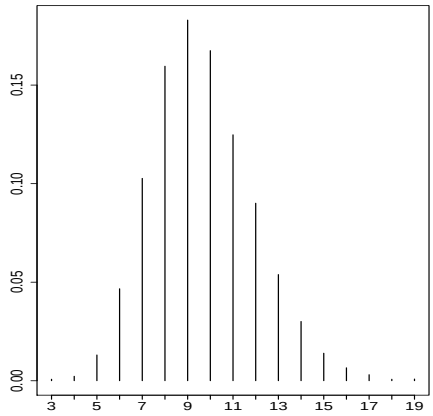
In the experiments I we put a non informative prior for κ, a Gamma distribution with mean 1 and variance 0.5 and we let σ vary over a grid on the interval $(0, 1)$. The Figure 3.26 shows an almost linear relation between



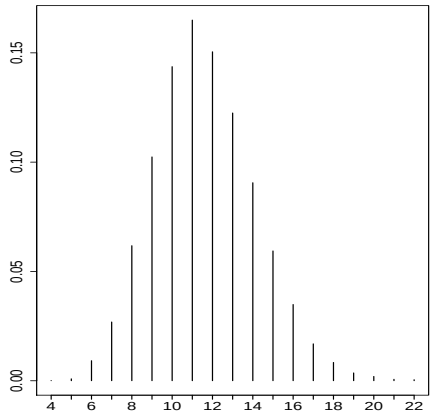
(a) Test L0: the posterior mean number of groups is 8.42.



(b) Test L1: the posterior mean number of groups is 9.06.

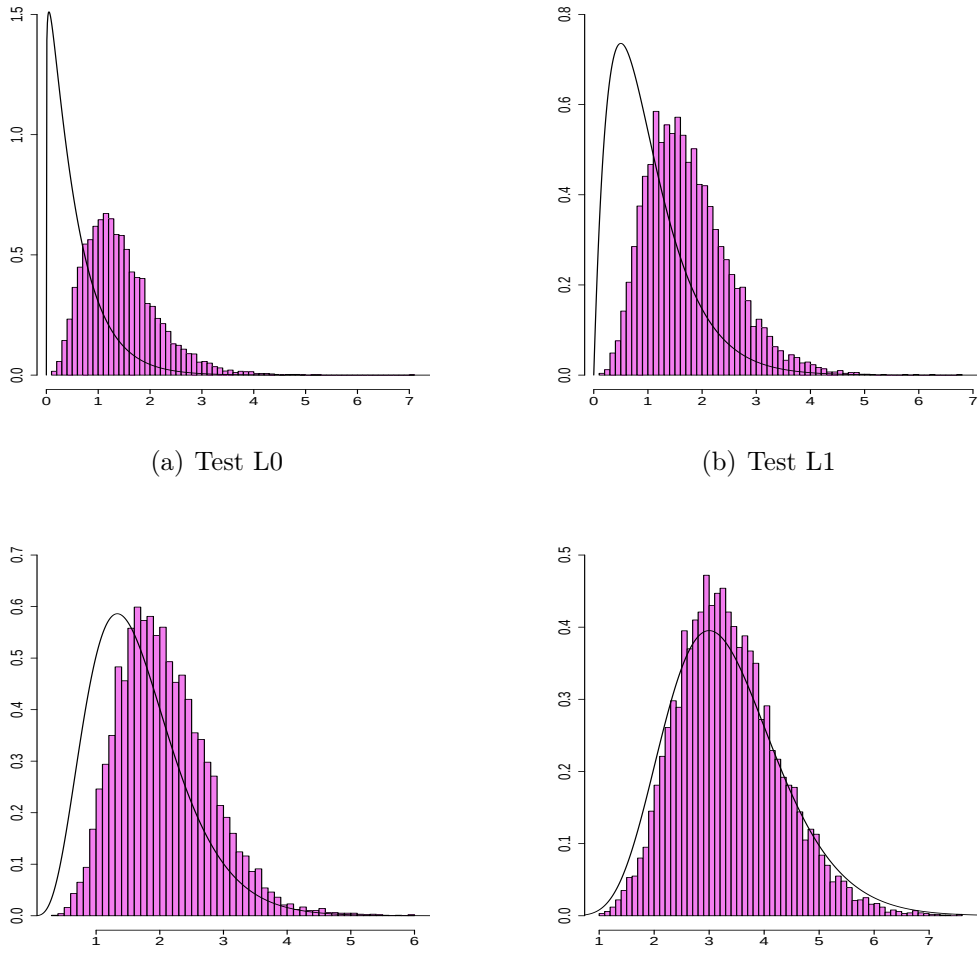


(c) Test L2: the posterior mean number of groups is 9.65.



(d) Test L3: the posterior mean number of groups is 11.6.

Figure 3.24: Histograms of the number of clusters K_n.



(a) Test L0

(b) Test L1

(c) Test L2

(d) Test L3

Figure 3.25: Histograms of parameter κ in tests of group L.

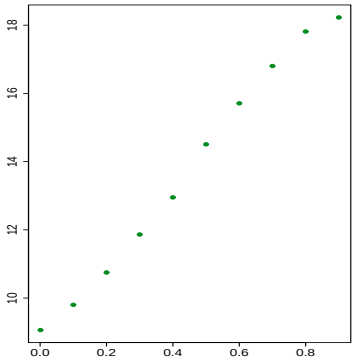
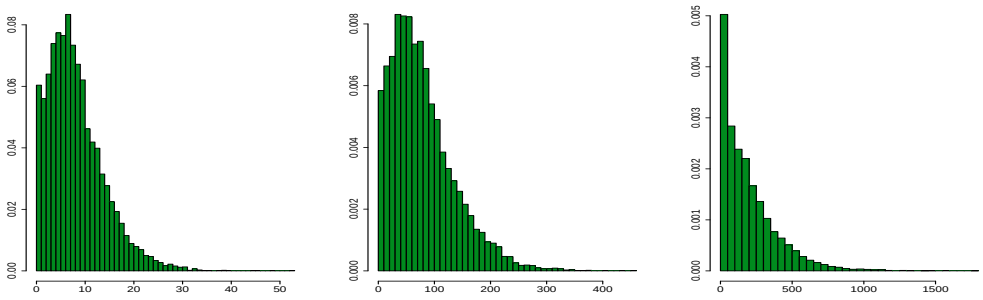


Figure 3.26: Values of the fixed parameter σ versus posterior mean number of clusters, $\mathbb{E}(K_n|data)$, in test I.

σ and the posterior number of clusters K_n.

There is a general increase in the number of non allocated jumps: this is due most of all to the augment of the parameter σ. This is clear looking at Figure 3.27.

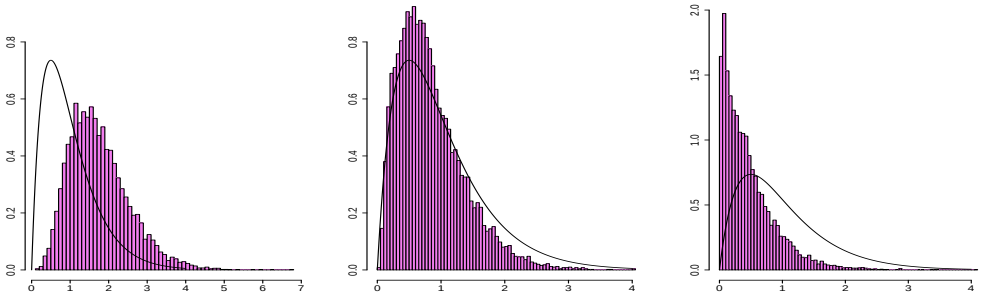


(a) Test I0: $\sigma = 0.001$ (b) Test I5: $\sigma = 0.5$ (c) Test I9: $\sigma = 0.9$

Figure 3.27: Histograms of the number of non-allocated jumps in some tests of group I: looking at the scale along the x-axis it is clear the increase of the number of jumps with σ.

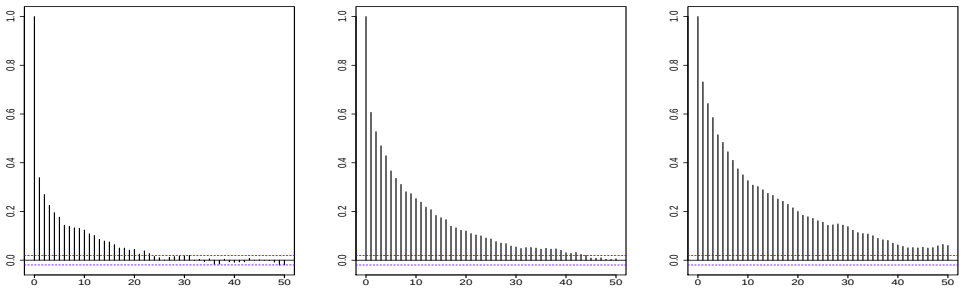
Observing the histograms of κ in Figure 3.28 we see the attempt of the parameter to balance the effect of the parameter σ.

If σ is small, κ tends to assume larger values with respect to the prior distribution, as in case I0; if σ is too large, the posterior distribution of κ shifts towards smaller values because there is no need to have a richer model, as in case I9.



(a) Test I0: $\sigma = 0.001$. (b) Test I5: $\sigma = 0.5$. (c) Test I9: $\sigma = 0.9$.

Figure 3.28: Histograms of κ.



(a) Test I0: $\sigma = 0.001$. (b) Test I5: $\sigma = 0.5$. (c) Test I9: $\sigma = 0.9$.

Figure 3.29: Autocorrelation of the variable κ.

The autocorrelation of the variable κ increases when σ assumes large values: see Figure 3.29. In general, with large values of σ some numerical problems appear due to the Incomplete Gamma function. It would be necessary to use a library with an higher precision, for example PARI.

3.3.4 | When σ and κ are both random

We now introduce some tests where both σ and κ are considered (independent) random variables.

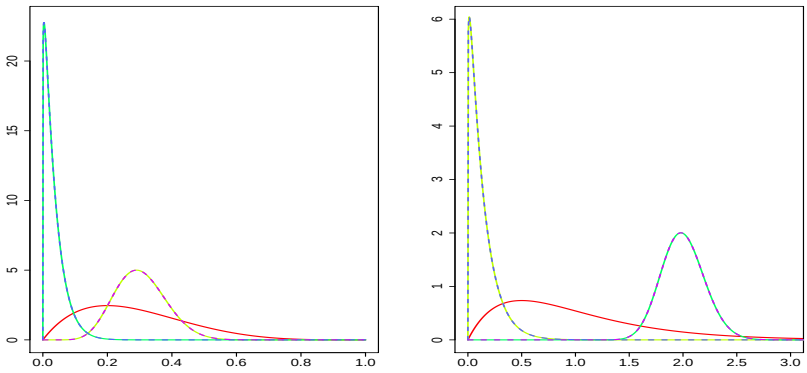
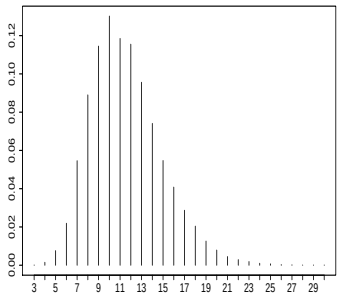


Figure 3.30: Five different couples of priors used in experiments of group M for σ (left) and κ (right), as in Table 3.3. Every color specifies a different couple.

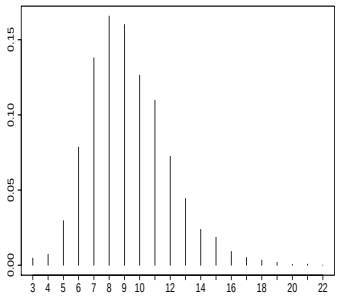
In the experiment M we choose five different couple of priors: they are represented in Figure 3.30, where each couple is identified by a color. In particular, we give as prior various information about the distribution of the two parameters and, consequently, of the number of clusters: a diffused distribution for both the parameters (non informative, as in M0, red), conflicting believes (as in case M1, yellow, where the prior for σ gives great mass on relatively large values while the distribution of κ is concentrated over very small values and M2, green, where we have the opposite situation) or in accordance (as in cases M3, blue, where both the priors give a big mass to small values of the parameters and M4, purple, where there is a complementary condition).

Remember that choosing a very small ε, as in this case, means consider a process that approximates the NGG process.

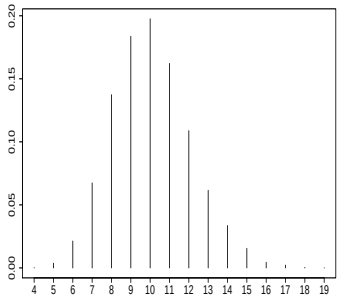
Looking at the histograms of K_n, we deduce from the first test, where the prior information was vague, that the model guesses the groups are about



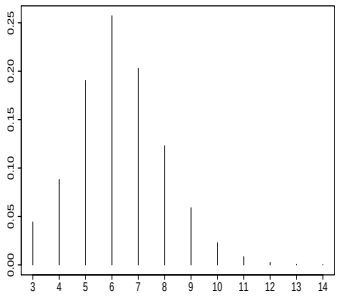
(a) Test M0



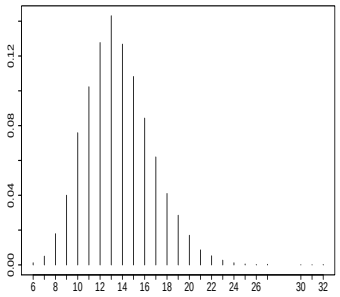
(b) Test M1



(c) Test M2

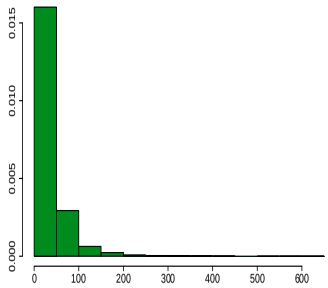


(d) Test M3

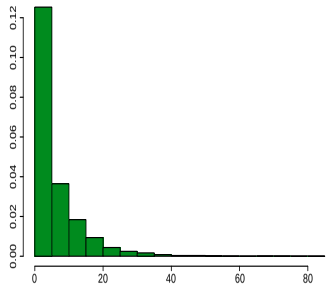


(e) Test M4

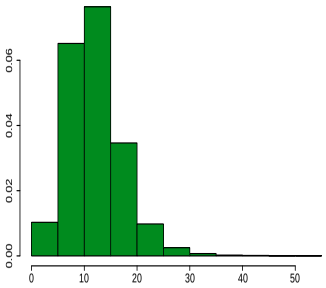
Figure 3.31: Histograms of K_n in tests M.



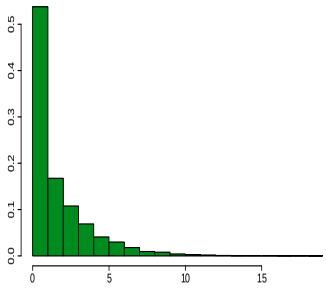
(a) Test M0



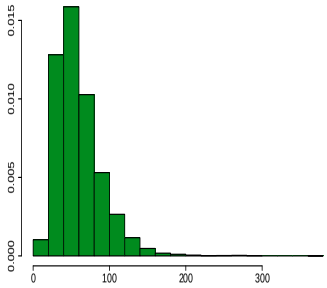
(b) Test M1



(c) Test M2



(d) Test M3



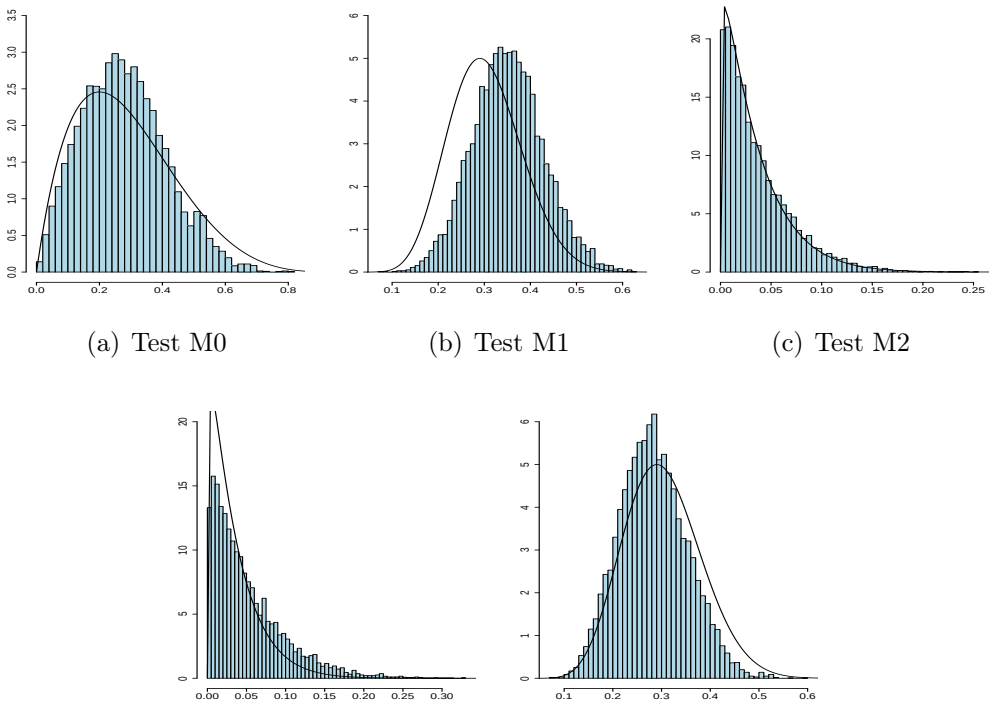
(e) Test M4

Figure 3.32: Histograms of N_{na} in tests M.

11. This is a large value: the model has to be coerced with the prior belief to consider less groups otherwise it identifies too many clusters in the data. The density estimate in any case is good.

Obviously, if the prior information on the 2 parameters is in agreement, then the posterior $\mathbb{E}(K_n)$ will be large or small accordingly. The mean number of groups is about 10 if the prior information on σ and κ is in disagreement (look at Figure 3.31).

Furthermore, observing in Figure 3.32 the histograms of the non-allocated jumps, we notice that they are a lot in the non-informative case M0 and if the two parameters assume large values, as in M4.



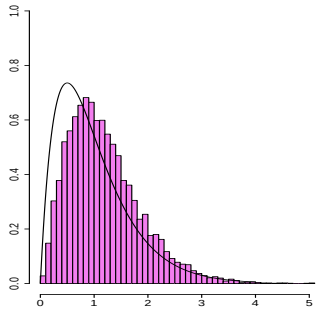
(a) Test M0 (b) Test M1 (c) Test M2

(d) Test M3 (e) Test M4

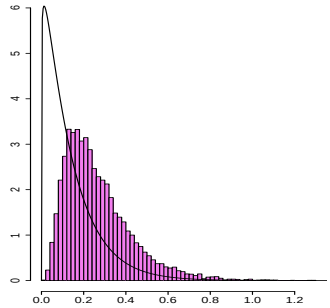
Figure 3.33: Histograms of the variable σ in tests M.

From cases M1 and M3 we deduce that σ influences the variance of the distribution of the variable N_{na}, in fact if σ is big the tails of the distribution are heavier.

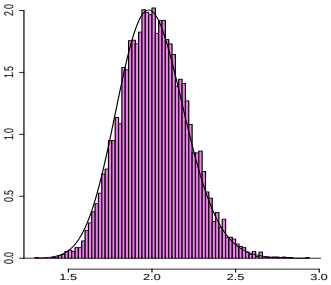
Another interesting issue is the behavior of the posterior chains of σ and κ: as we can observe from Figures 3.33 and 3.34, they are quite faithful to their prior distributions, even if in general the variable κ is more flexible like, for example, in test M3 where both the priors are concentrated on very small



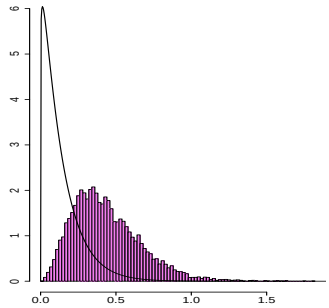
(a) Test M0



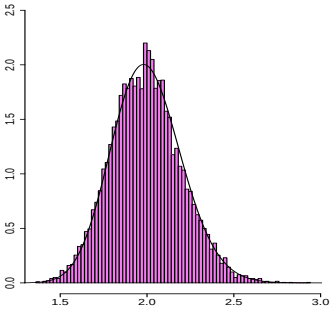
(b) Test M1



(c) Test M2



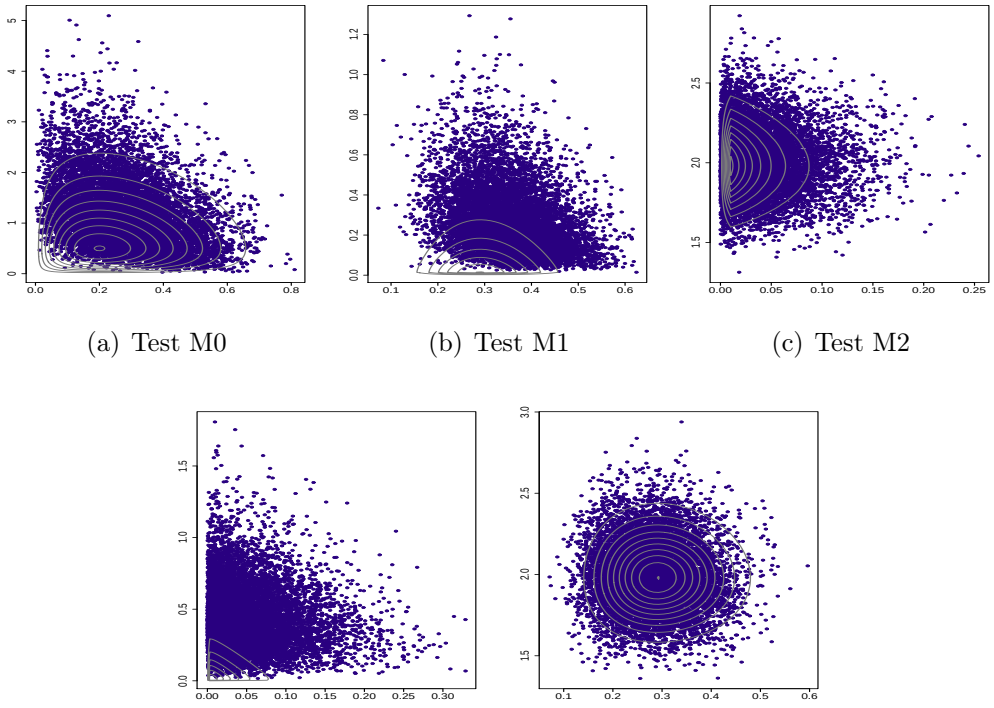
(d) Test M3



(e) Test M4

Figure 3.34: Histograms of the variable κ in tests M.

values and the parameter κ is shifted towards larger values while σ remains faith to the prior distribution.



(a) Test M0 (b) Test M1 (c) Test M2

(d) Test M3 (e) Test M4

Figure 3.35: Scatterplots of σ versus κ: in gray the contour levels of the conjugate prior distribution over the couple (σ, κ).

Figure 3.35 shows the scatterplot of σ versus κ in tests M: the contour levels of the prior distribution are superimposed. We notice the shifting of the points from the prior distribution in cases M1 and M3 where the prior model is too restrictive on the values of the parameters.

Chapter 4 | Yeast cell cycle data

In this final chapter we apply our model to a multivariate dataset, the Yeast cell cycle data. First, a robustness analysis is performed, as we did in Chapter 3 for the unidimensional case; then, we use the model for cluster analysis. Indeed, since we are dealing with a gene expression dataset, this application can be very interesting: clustering techniques have proven to be helpful to understand gene functioning, gene regulation and cellular processes. Genes with similar expression patterns can be clustered together, hence suggesting similar cellular functions.

4.1 | Description of the robustness analysis

We provide in this section a robustness analysis in a multidimensional case: in particular, we fitted our model to a dataset used in the literature for clustering gene expression profiles, usually called Yeast cell cycle data, and represented in Figure 4.1. We are considering a gene expression dataset from a microarray experiment: a microarray is an array of DNA molecules that permits many hybridization experiments to be performed in parallel. It can monitor expression levels of thousands of genes simultaneously in multiple conditions (in this case we obtain a time-series during a biological process). The dataset can be represented by a real-valued matrix $[X_{ij},\ 1 \leq i \leq n,\ 1 \leq j \leq p]$, where the rows $(X_1, \ldots, X_n)$ contain the expression patterns of genes and will be our data points. Each cell X_{ij} is the measured expression level of gene i at time j. The Yeast cell cycle dataset contains $n = 389$ gene expression profiles, observed at 17 different time values, one every 10 minutes from time zero. We chose a subset of the original dataset, representing the second cell cycle. The final dataset ($n = 389$, $p = 9$) has been obtained standardizing each row of the gene expression matrix to have zero mean and unitary variance.

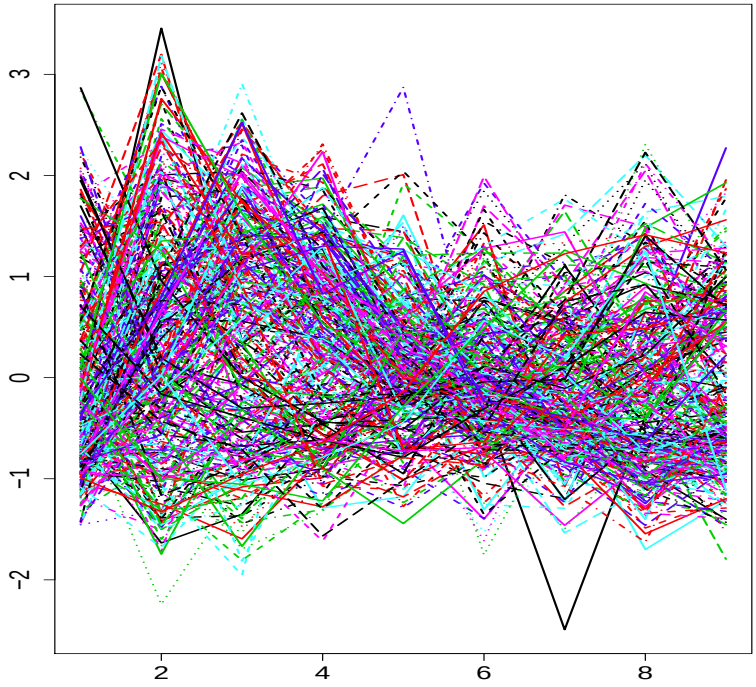


Figure 4.1: Yeast cell cycle data: on the x-axis the 9 time steps in which the 389 gene expression profiles are observed. On the y-axis the measured expression level of the n different genes after the standardization.

We model data as follows:

$$\begin{cases} X_i|\theta_i \overset{ind}{\sim} N_p(\cdot|\theta_i) \quad i = 1, .., n \\ \theta_1, \theta_2, \ldots, \theta_n|P_\varepsilon \overset{iid}{\sim} P_\varepsilon \\ P_\varepsilon \sim \varepsilon - NGG(\sigma, \omega, \kappa, P_0) \end{cases}$$

where $\theta_i = (\mu_i, \Sigma_i)$ and Σ_i, the covariance matrix, is assumed diagonal, as follows:

$$\begin{pmatrix} \sigma^2_{1,i} & 0 & \ldots & 0 \\ 0 & \sigma^2_{2,i} & \ldots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \ldots & \sigma^2_{p,i} \end{pmatrix}$$

In this way the correlation between the variables is ignored, but we will see that the model is enough flexible. We fixed P_0 as:

$$N_p\left(\mu_i|m_0, \frac{1}{s_0}\Sigma_i\right) \times \prod_{k=1}^{p} IG(\sigma^2_{k,i}|a, b).$$

As far as the choice of hyperparameters for P_0 is concerned, it is straightforward to see that

- $\mathbb{E}(\mu) = m_0$
- $Var(\mu) = \dfrac{b}{(a-1)s_0}\mathbb{I}_p$
- $\mathbb{E}(\Sigma) = \dfrac{b}{(a-1)}\mathbb{I}_p$
- $Var(\sigma^2_1, \ldots, \sigma^2_p) = \dfrac{b^2}{(a-1)^2(a-2)}\mathbb{I}_p$

where $\mathbb{I}_p$ is the identity matrix of dimension $p \times p$. For our analysis we selected two different sets of hyperparameters for P_0: the first is $(m_0, s_0, a, b) = (\mathbf{0}, 1, 2.1, 0.11)$ such that $Var(\mu) = 0.1\mathbb{I}_p$, $\mathbb{E}(\Sigma) = 0.1\mathbb{I}_p$ and $Var(\sigma^2_i) = 0.1$ for every i, while the second is $(\mathbf{0}, 1, 3, 2)$ in order to have $Var(\mu) = \mathbb{I}_p$, $\mathbb{E}(\Sigma) = \mathbb{I}_p$ and $Var(\sigma^2_i) = 1$ for every i.

Name	ε	(m_0, s_0, a, b)	σ	κ
A0, ..., A3	10^{-6}	$(\mathbf{0}, 1, 2.1, 0.11)$	$\{0.001, 0.1, 0.2, 0.5\}$	0.7
a0, ..., a3	10^{-6}	$(\mathbf{0}, 1, 3, 2)$	$\{0.001, 0.1, 0.2, 0.5\}$	0.7
B0, ..., B3	$Unif(0, 0.01)$	$(\mathbf{0}, 1, 2.1, 0.11)$	$\{0.001, 0.1, 0.2, 0.5\}$	0.7
b0, ..., b3	$Unif(0, 0.01)$	$(\mathbf{0}, 1, 3, 2)$	$\{0.001, 0.1, 0.2, 0.5\}$	0.7
C0, ..., C3	10^{-4}	$(\mathbf{0}, 1, 2.1, 0.11)$	$Beta(2, 15)$	$\{0.1, 1, 3, 10\}$
c0, ..., c3	10^{-4}	$(\mathbf{0}, 1, 3, 2)$	$Beta(2, 15)$	$\{0.1, 1, 3, 10\}$
D0, ..., D3	10^{-4}	$(\mathbf{0}, 1, 2.1, 0.11)$	$\{0.001, 0.1, 0.2, 0.5\}$	$Gamma(2, 0.1)$
d0, ..., d3	10^{-4}	$(\mathbf{0}, 1, 3, 2)$	$\{0.001, 0.1, 0.2, 0.5\}$	$Gamma(2, 0.1)$
E0	10^{-4}	$(\mathbf{0}, 1, 2.1, 0.11)$	$Beta(2, 15)$	$Gamma(2, 0.1)$
e0	10^{-4}	$(\mathbf{0}, 1, 3, 2)$	$Beta(2, 15)$	$Gamma(2, 0.1)$
F0, ..., F3	$\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$	$(\mathbf{0}, 1, 2.1, 0.11)$	0.001	0.7
f0, ..., f3	$\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$	$(\mathbf{0}, 1, 3, 2)$	0.001	0.7
G0, ..., G2	10^{-4}	$(\mathbf{0}, 1, 2.1, 0.11)$	$\{0.001, 0.1, 0.2\}$	$\{.7, .4, .1\}$
g0, ..., g2	10^{-4}	$(\mathbf{0}, 1, 3, 2)$	$\{0.001, 0.1, 0.2\}$	$\{.7, .4, .1\}$

Table 4.1: Scheme of the tests in the robustness analysis. In these experiments, the final sample size produced by the algorithm is 5000 iterations, after a burnin period of 5000 and a thinning of 20.

Note that in the experiments A, a, B, b, F and f we have fixed $\kappa = 0.7$ so that when $\sigma = 0.001$ $\mathbb{E}(K_n)$ is equal to 5, that is the number of clusters identified by Cho et al. (1998) by visual inspection. Remember that K_n is the variable number of groups in the data. We will return on this aspect later. Furthermore, in the tests B the prior on the variable ε is concentrated over smaller values with respect to the unidimensional case. This is because, in the multivariate case, the MCMC algorithm is more complex and the model is more sensible with respect to hyperparameters' choice. We bumped into the following problem: large values of ε yield parametric models and in this case the MCMC chains of N_ε and K_n assume always the same, fixed, value.

In tests e0 and E0 we consider both σ and κ random: the prior on σ is a Beta distribution that gives most mass between 0 and 0.3, while the prior for κ is a Gamma with mean 20 and very large variance in order to be non-informative.

On the other hand, in tests G and g we chose 3 different couples of values for (σ, κ) such that $\mathbb{E}(K_n) = 5$ and the variance, as we have seen also in Chapter 3, goes up with σ. The corresponding prior distributions for the variable K_n in the 3 cases are reported in Figure 4.2.

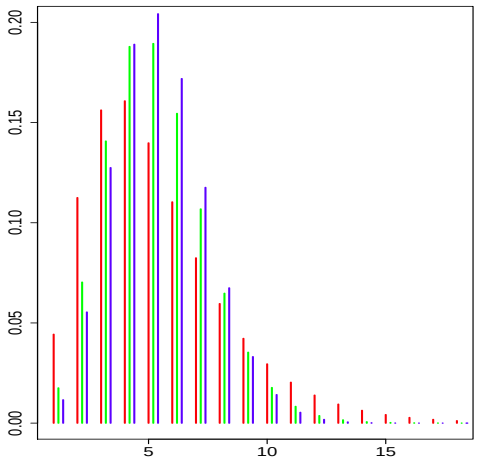


Figure 4.2: Prior distributions of the variable K_n in tests G0 with $(\sigma, \kappa) = (0.001, 0.7)$ (blue), G1 where $(\sigma, \kappa) = (0.1, 0.4)$ (green) and G2 with $(\sigma, \kappa) = (0.2, 0.1)$ (red). All the prior distributions have mean equal to 5, while the variance is larger as σ gets bigger, as in G2.

Before presenting the Bayesian inference we got, we introduce an index that is useful for evaluating the performance of the model and compare the

different sets of parameters: the Log Pseudo Marginal Likelihood (LPML) that is

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

where the CPO_i is the conditional predictive ordinate of the i-th sample, X_i. The CPO is the value of the predictive distribution evaluated at X_i conditioning on the training sample $\mathbf{X}_{-i}$:

$$CPO_i = f_i(X_i|\mathbf{X}_{-i}), \quad i = 1, .., n.$$

Obviously if the values of the CPO (and of the LPML, of course) are large the model fits the data well.

In order to evaluate these indexes, the following formulas make the computation simpler, taking advantage of the MCMC algorithm we built in Chapter 2:

$$\begin{aligned} CPO_i &= \int_\Theta f_i(X_i|\theta, \mathbf{X}_{-i})\mathcal{L}(\theta|\mathbf{X}_{-i})d\theta = \int_\Theta f_i(X_i|\theta)\frac{\prod_{j\neq i} f_j(X_j|\theta)\pi(d\theta)}{\int_\Theta \prod_{j\neq i} f_j(X_j|\theta)\pi(d\theta)} \\ &= \int_\Theta \frac{\prod_{j=1}^{n} f_j(X_j|\theta)\pi(d\theta)}{\int_\Theta \prod_{j\neq i} f_j(X_j|\theta)\pi(d\theta)} \end{aligned}$$

where we used the Bayes' theorem. We obtain:

$$\begin{aligned} \frac{1}{CPO_i} &= \frac{\int_\Theta \prod_{j\neq i} f_j(X_j|\theta)\pi(\theta)d\theta}{\int_\Theta \prod_{j=1}^{n} f_j(X_j|\theta)\pi(\theta)d\theta} = \int_\Theta \frac{1}{f_i(X_i|\theta)}\frac{\prod_{j=1}^{n} f_j(X_j|\theta)\pi(d\theta)}{\int_\Theta \prod_{j=1}^{n} f_j(X_j|\theta)\pi(d\theta)} \\ &= \int_\Theta \frac{1}{f_i(X_i|\theta)}\pi(\theta|\mathbf{X}) \simeq \frac{1}{G}\sum_{g=1}^{G} \frac{1}{f_i(X_i|\theta^{(g)})}, \end{aligned}$$

where G is the number of total iterations and $\theta^{(g)}$ is the value of the chain at iteration g. In Table 4.2, values of the LPML index for every test are reported. However, observe that a more complex model will usually be able to explain the data better, and consequently will yield a higher LPML. In fact, it is clear from the table that this index depends strongly from the choice of P_0: when the hyperparameters in P_0 are $(\mathbf{0}, 1, 2.1, 0.11)$, all the test experiments provide more groups to describe the data. As expected, this yields larger values of LPML for all the experiments named with the capital letters.

Let us see some general comments on density estimation: since all the estimates are similar we will present here only few density estimates.

Name	LPML
A0, ..., A3	$\{5.2, 5.04, 5.08, 4.96\}$
a0, ..., a3	$\{1.63, 1.63, 1.61, 1.56\}$
B0, ..., B3	$\{5.09, 5.2, 5.15, 5.1\}$
b0, ..., b3	$\{1.67, 1.68, 1.69, 1.71\}$
C0, ..., C3	$\{5, 5.15, 5.09, 5.08\}$
c0, ..., c3	$\{1.67, 1.68, 1.69, 1.71\}$
D0, ..., D3	$\{5.14, 5.09, 5.12, 5.2\}$
d0, ..., d3	$\{1.63, 1.63, 1.65, 1.69\}$
E0	$\{5.12\}$
e0	$\{1.63\}$
F0, ..., F3	$\{5.18, 5.03, 5.04, 5.19\}$
f0, ..., f3	$\{1.63, 1.64, 1.63, 1.65\}$
G0, ..., G2	$\{5.04, 5.15, 5.2\}$
g0, ..., g2	$\{1.63, 1.63, 1.67\}$

Table 4.2: Values of LPML index for every test experiment.

As we have just seen from the analysis of LPML, what actually affects the estimation is the choice of P_0 in the ε-NGG. Figures 4.3 and 4.4 show the Bayesian density estimates, i.e. the marginal predictive densities along the different p directions; in Figure 4.3 all the parameters are fixed while in Figure 4.4 σ and κ are random. It is obvious that changing P_0, the fitting of the model to the data changes. In the tests with the capital letters (the green curves in Figures 4.3 and 4.4) the model needs more groups to describe the data. The corresponding estimate is closer to the histogram of the data with respect to the tests with the lower case letter (a0 and e0 in blue in Figures 4.3 and 4.4).

We do not observe substantial differences in the density estimation plots in Figures 4.3 and 4.4, thus implying the robustness of the model with respect to the choice of σ and κ.

Consider now test B0, where ε is a random variable with a Uniform prior: also in this case the estimates are good. In Figure 4.5 we report the density estimates. The points with a low CPO (lower than the 10% observed quantile) are shown in red.

Finally, Figure 4.6 shows the marginal (bi-dimensional) predictive distributions for test experiments a0 and A0. The results are encouraging because, although the true distribution of the data is not available, we observe a good fit between the points and the marginal predictive density. However, observe that the figure concerns only a 2D slice of a 9 dimensional space, and when we display this marginal (bivariate) predictive density, we do not show any statistical phenomenon which might occur in the larger space.

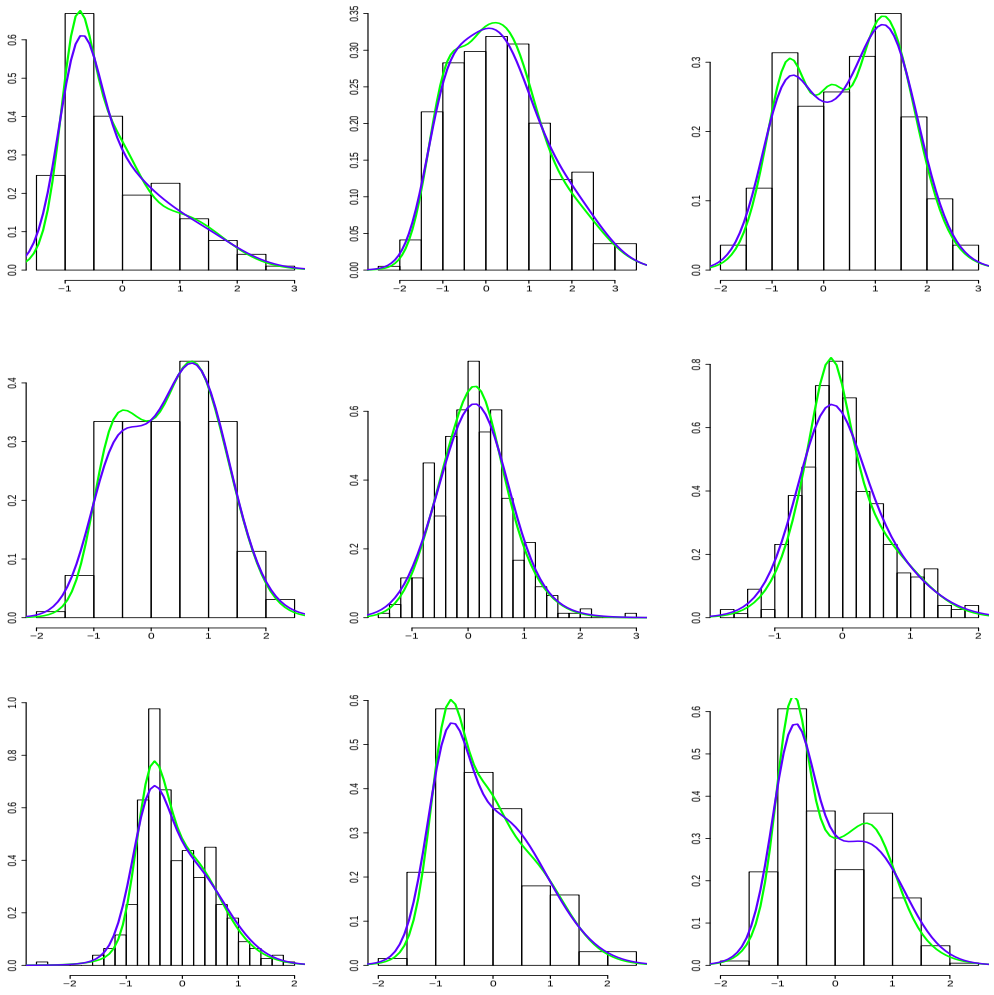


Figure 4.3: Marginal density estimates along the 9 directions for the test A0 (green) and a0 (blue).

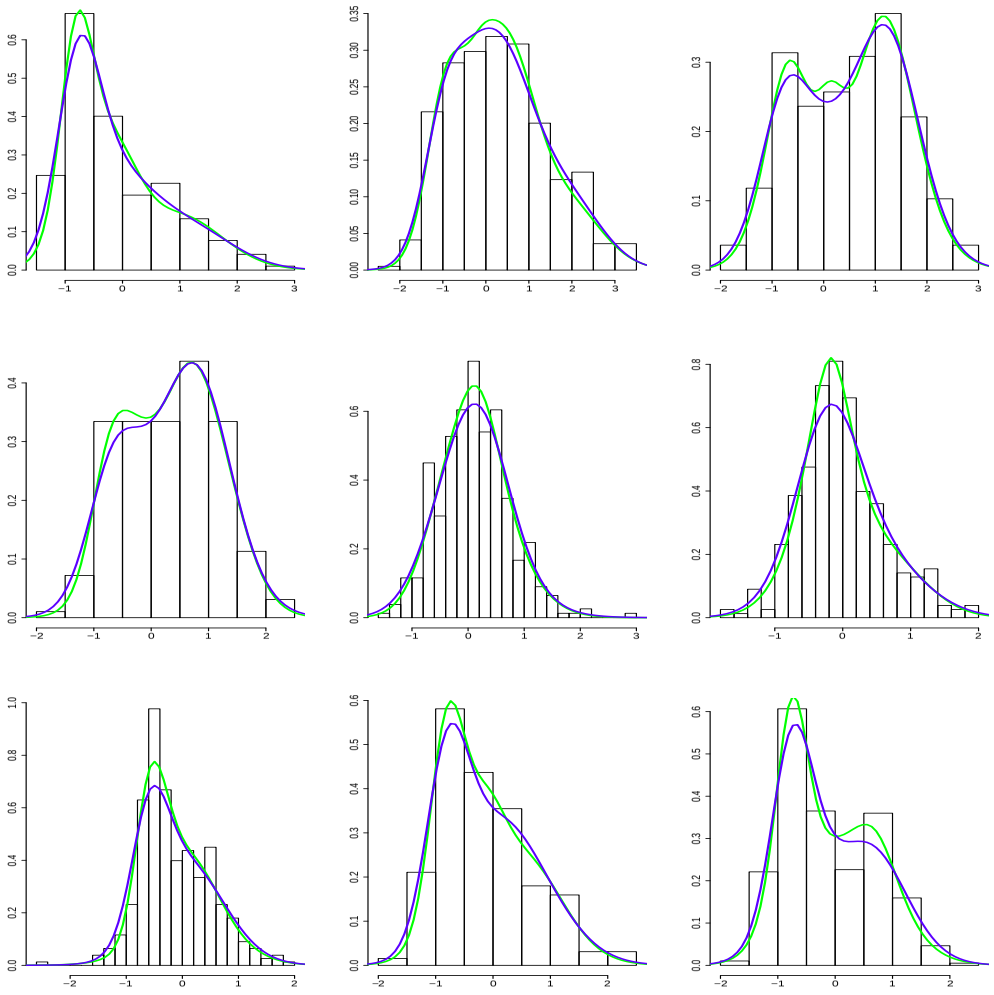


Figure 4.4: Marginal density estimates along the 9 directions for the test E0 (green) and e0 (blue).

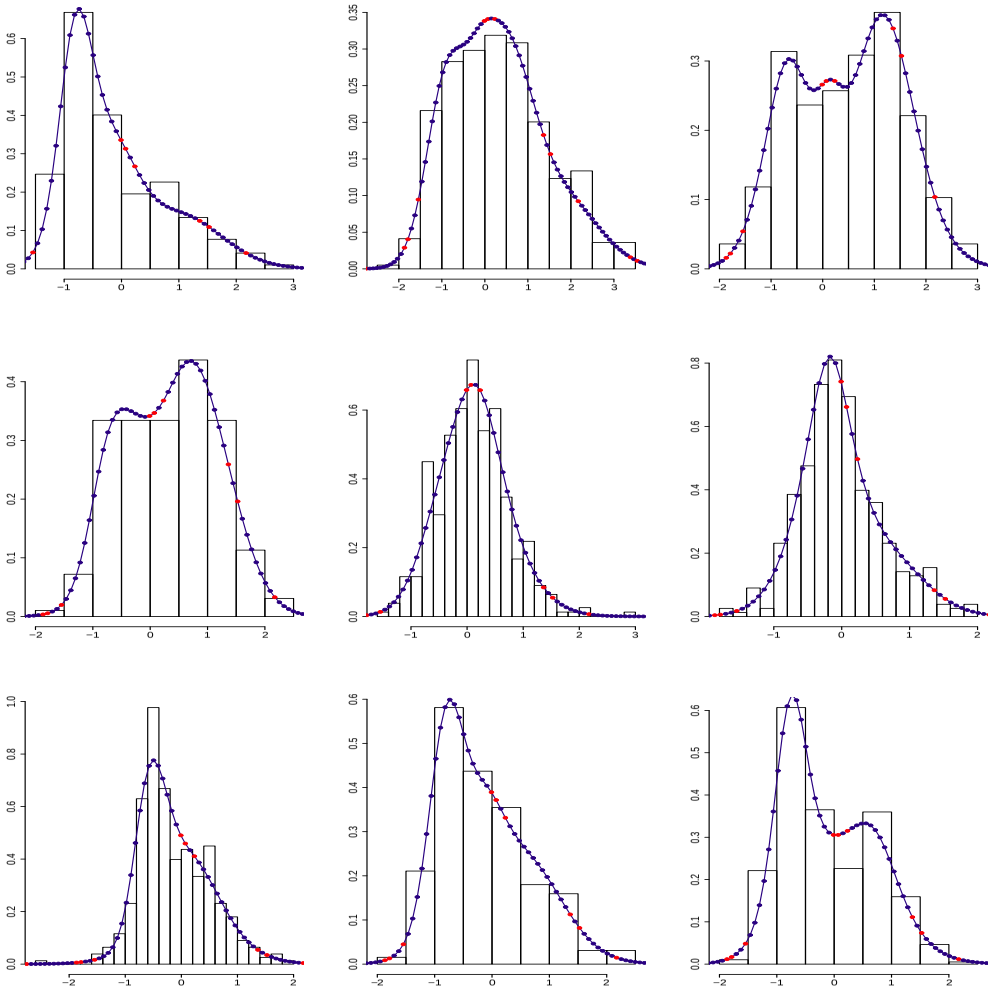


Figure 4.5: Marginal density estimates along the 9 directions for the test B0; data for which the CPO is lower than the 10% observed CPO quantile are depicted in red.

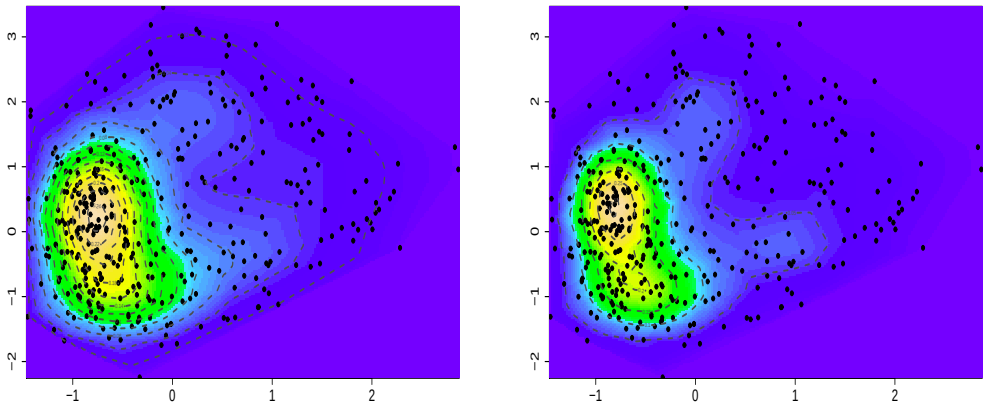


Figure 4.6: Bivariate density estimates (along the first 2 dimensions) for tests a0 (left) and A0 (right).

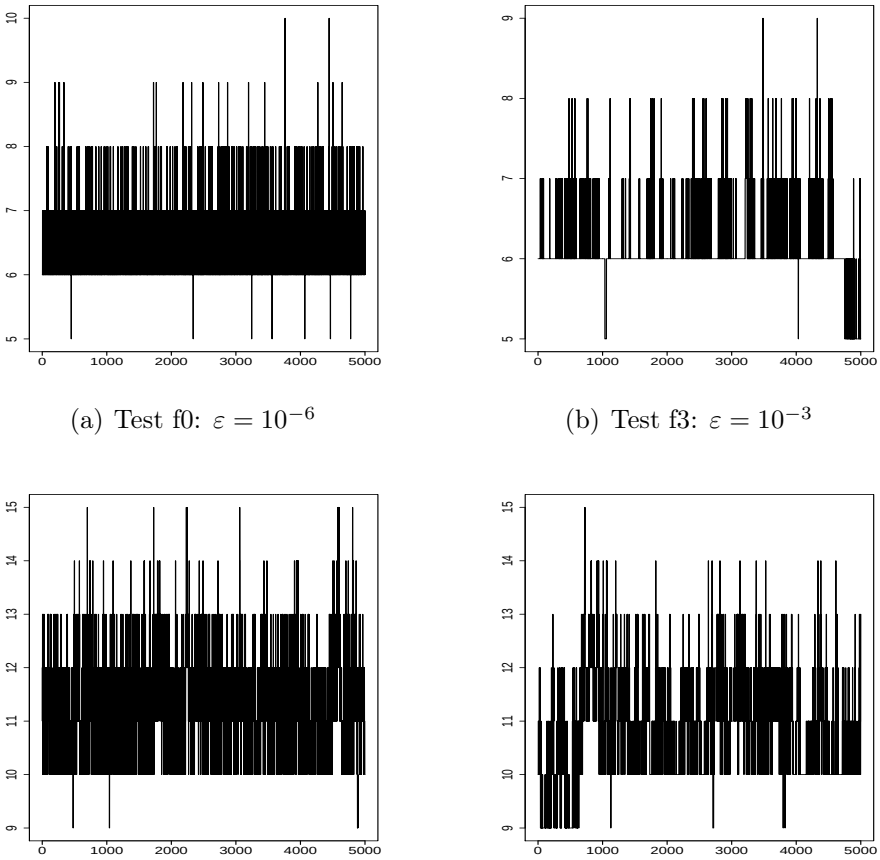
4.1.1 | The ε-NGG mixture model with fixed parameters

We report in this section some results concerning the experiments where all the parameters are kept fixed in order to investigate their effects on the posterior estimates. In particular, we are going to comment the inference for tests F and f, where the parameter ε increases, and the sets G and g, where different couples (σ, κ) are considered.

The first aspect to notice in tests F (and f) is that the posterior chains of the number of clusters K_n and of the total number of components of the mixture $N_\varepsilon + 1$ get worse if ε increases (see Figure 4.7). In the multivariate case the model is very sensitive with respect to ε and it is necessary to keep it low in order to obtain good posterior chains. In fact, if ε is relatively large, the jumps J are sampled from a distribution which is almost a Dirac's delta in ε, and therefore they will assume the same value. For this reason we will obtain $N_\varepsilon + 1 = K_n$ constant.

The number of non-allocated jumps is larger when ε is smaller: obviously the run-time of the algorithm will be greater but the mixing will be better. In particular, a larger number of small jumps are included in the process when ε is small: they have an important role in the multivariate case, since increasing ε yields "bad" posterior chains and a very slow mixing.

Consider now tests G and g: we have already seen in Figure 4.2 how the prior distribution of the number of groups changes in the tests.
When σ is relatively large and, of course, κ becomes small, in order to keep



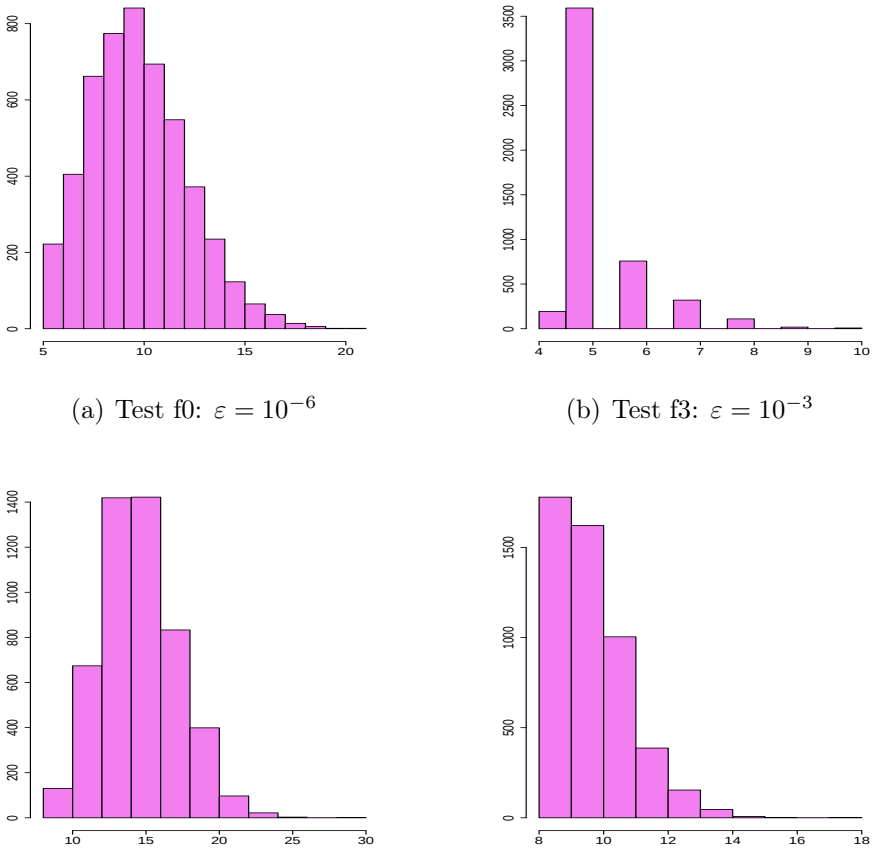
(a) Test f0: $\varepsilon = 10^{-6}$

(b) Test f3: $\varepsilon = 10^{-3}$

(c) Test F0: $\varepsilon = 10^{-6}$

(d) Test F3: $\varepsilon = 10^{-3}$

Figure 4.7: Traceplots of K_n, the number of groups, in some tests of groups f and F.



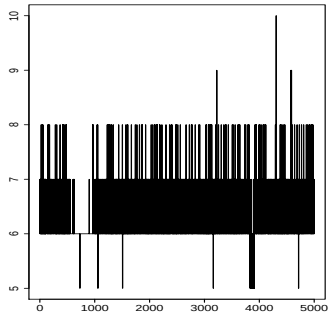
(a) Test f0: $\varepsilon = 10^{-6}$

(b) Test f3: $\varepsilon = 10^{-3}$

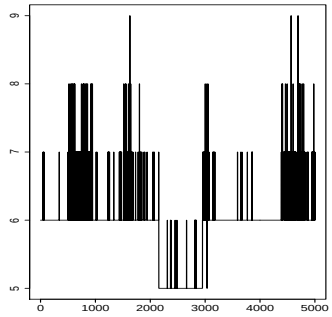
(c) Test F0: $\varepsilon = 10^{-6}$

(d) Test F3: $\varepsilon = 10^{-3}$

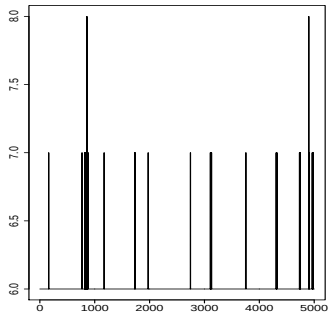
Figure 4.8: Posterior distribution (histograms of the MCMC draws) of the variable N_ε, where $N_\varepsilon + 1$ is the number of elements in the mixture, in some tests.



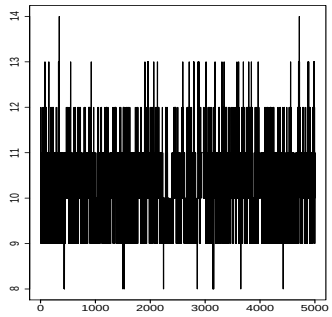
(a) g0



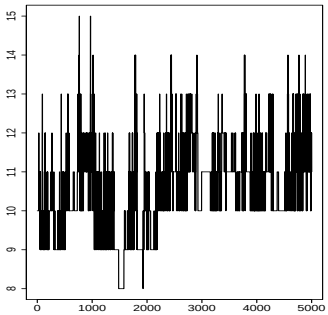
(b) g1



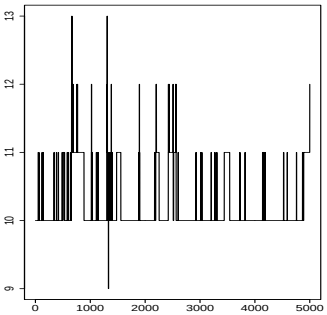
(c) g2



(d) G0

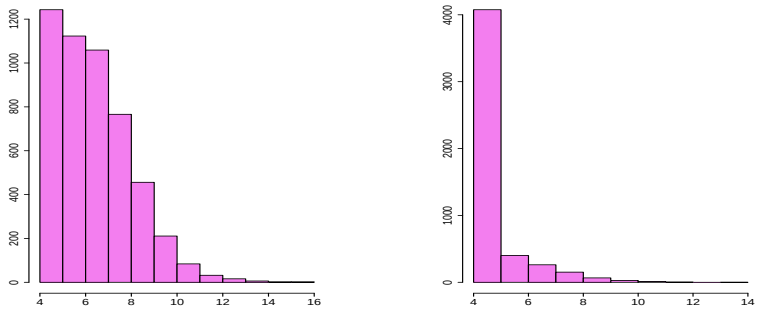


(e) G1



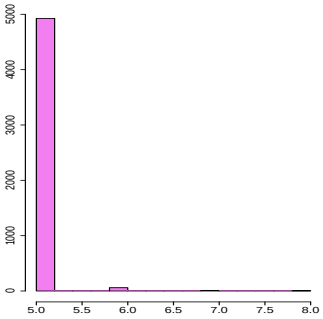
(f) G2

Figure 4.9: Traceplots of K_n in experiment tests G. It is clear that the mixing of the chain gets worse when σ becomes larger and κ smaller.

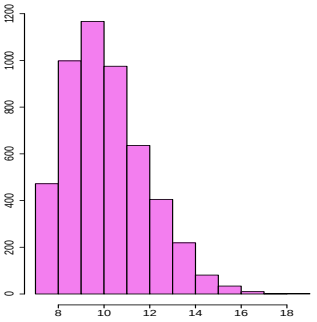


(a) g0

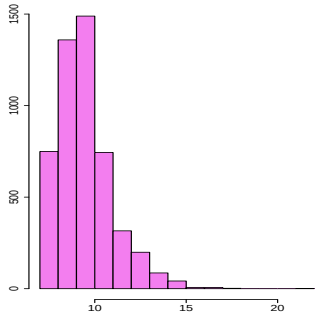
(b) g1



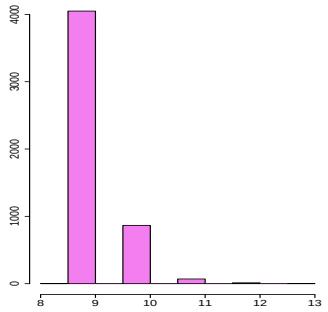
(c) g2



(d) G0



(e) G1



(f) G2

Figure 4.10: Posterior distributions (histograms of the MCMC draws) of the variable N_ε in some tests.

$\mathbb{E}(K_n)$ equal to 5, the posterior chains become "worse" as shown in Figure 4.9. The chains of K_n and N_ε get stuck: the model becomes "almost parametric" because the algorithm is not able to update the values of these two variables. In particular, when σ is large, the posterior variance of K_n becomes small (see Figure 4.10).

4.1.2 | Bayesian inference when ε is random

We focus here on tests a, A and b, B in order to compare the estimates with ε fixed and random. In the multivariate case the randomness of ε yields posterior chains with a bad mixing, in particular for the variable K_n, number of groups. Therefore, it is necessary to consider a suitable prior for ε, concentrated on very small values. In fact, in this way there is a gain in the computational time while the MCMC chains' mixing is still satisfactory.

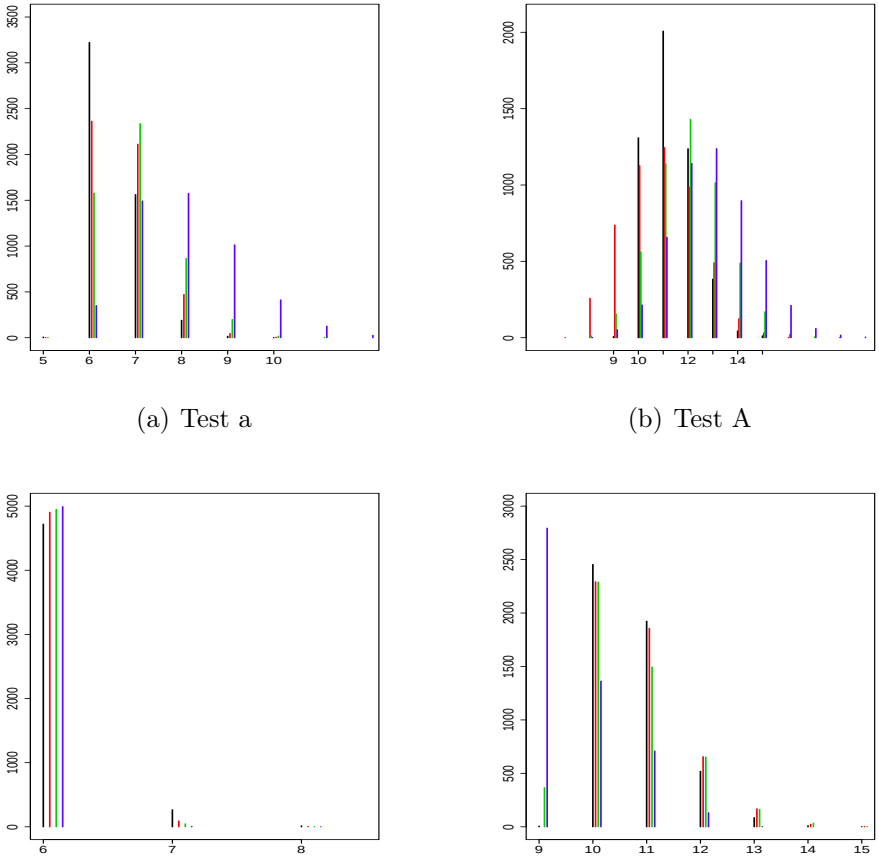
In Figure 4.11 posterior distributions of variable K_n, number of groups, are shown. In tests A and a it is clear how the distribution of the variable K_n is shifted towards larger values when σ increases (lines green and blue, corresponding to tests number 2 and 3) as in the unidimensional case. On the other hand, when ε is random (as in tests b and B), the posterior variance of K_n is very small and we can not appreciate the increase of the number of groups when σ goes up.

In tests A and a, the posterior chains have a good mixing both when σ is small or large (see Figure 4.12). We notice also a significant increase in N_ε and, consequently, of the non-allocated jumps when σ gets larger, as in the univariate case. On the contrary, in test experiments b and B, where ε is random, posterior chains have a bad mixing (in Figure 4.12 compare the test B0 and B3, for instance).

Another interesting aspect is the analysis of the posterior values assumed by ε. We can observe from Figure 4.13 that this variable assumes also relatively large values: when this occurs, the chains of the variables K_n and N_ε get stuck, leading to bad estimates. This phenomenon is more visible when σ is large, as in cases B3 and b3.

4.1.3 | Bayesian inference when both σ and κ are random

We now focus on tests where σ and κ are both random. In the test experiments C and c, we gave as prior for σ a $Beta(2, 15)$ distribution and we let κ vary over a grid from 0.1 to 10. Because of the randomness of σ, the number of groups K_n slightly depends on the choice of the parameters of the prior process: if κ is small, as in C0 and c0, the posterior chain of σ is shifted towards big values, while if κ is large, as in C3 and c3, the histogram will



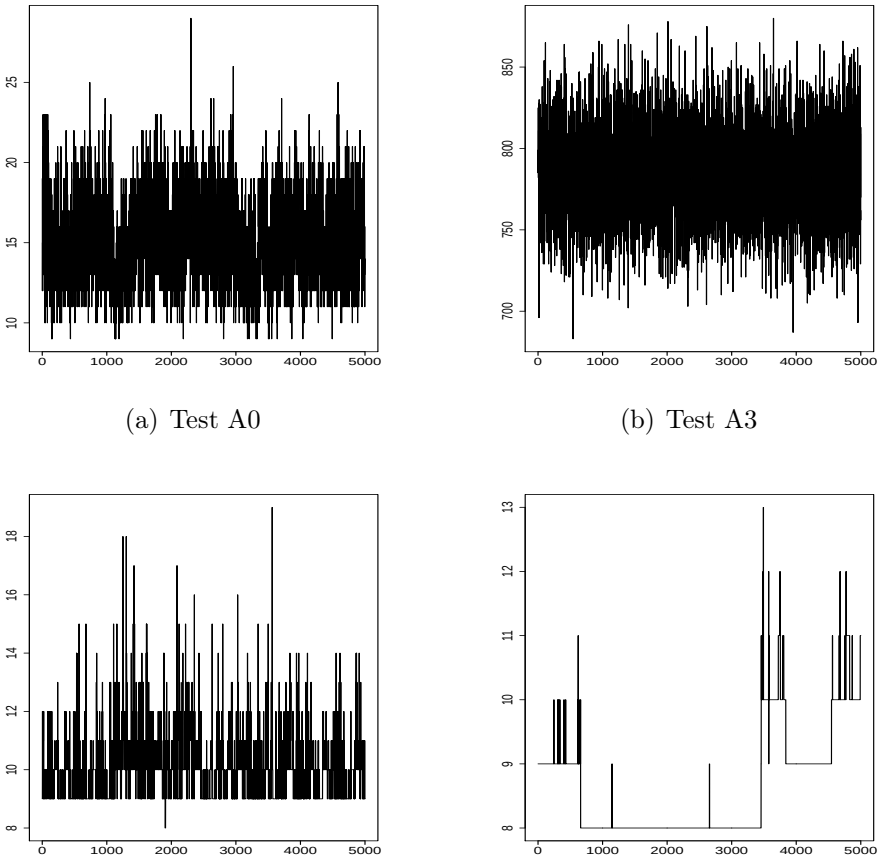
(a) Test a

(b) Test A

(c) Test b

(d) Test B

Figure 4.11: Here histograms of K_n, number of clusters, are superimposed in groups a, A, b and B. In black tests with σ equal to 0.001; in green with σ equal to 0.1, while red corresponds to $\sigma = 0.2$ and blue to $\sigma = 0.3$.



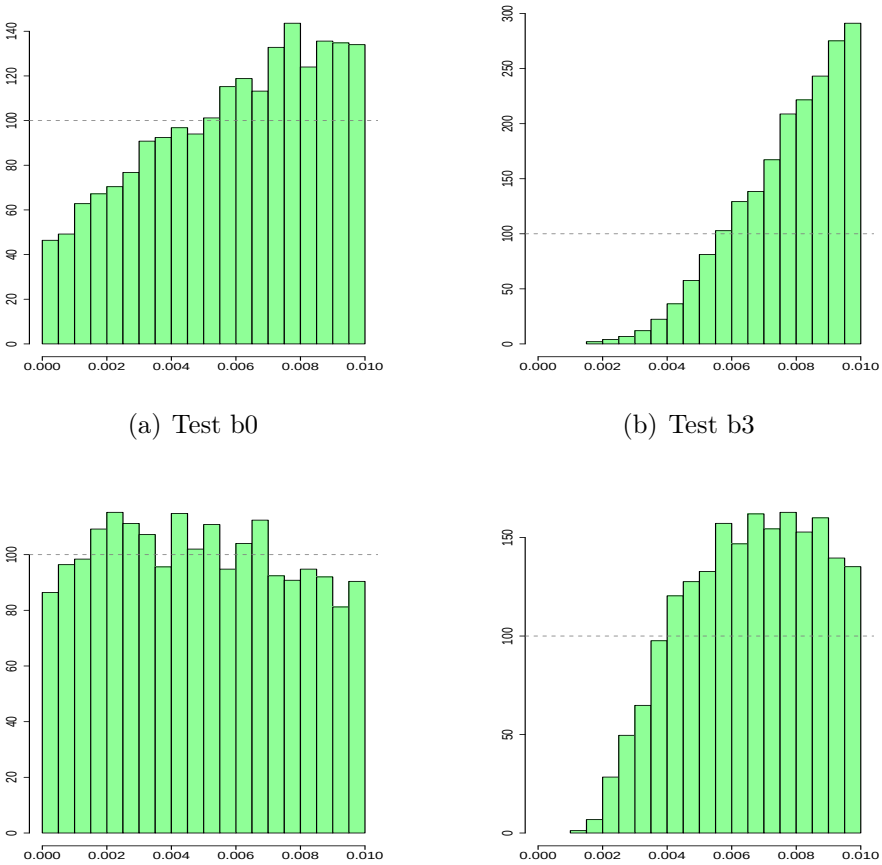
(a) Test A0

(b) Test A3

(c) Test B0

(d) Test B3

Figure 4.12: Traceplots of variable N_ε, where $N_\varepsilon + 1$ is the number of components of the mixture, in some tests.



(a) Test b0

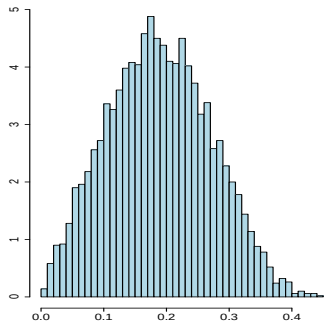
(b) Test b3

(c) Test B0

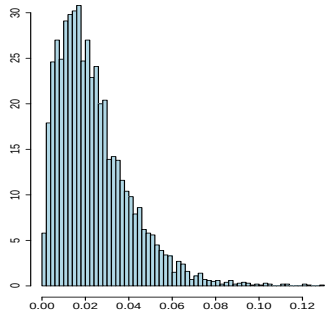
(d) Test B3

Figure 4.13: Histograms of the random variable ε. In gray the prior is represented: $Unif(0, 0.01)$.

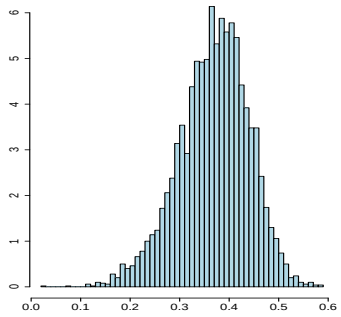
be concentrated over small values (compare Figure 4.14). In this sense, σ "balances" the value assumed by κ. From Figure 4.16 it is clear the shifting towards larger values of the posterior distribution of K_n when κ increases. As we just pointed out, this effect is weakened by the randomness of parameter σ.



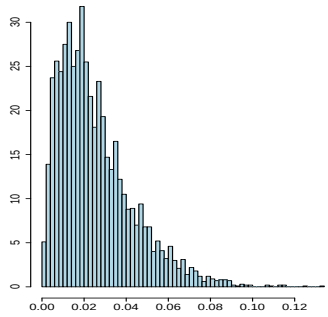
(a) Test c0: $\kappa = 0.1$



(b) Test c3: $\kappa = 10$



(c) Test C0: $\kappa = 0.1$

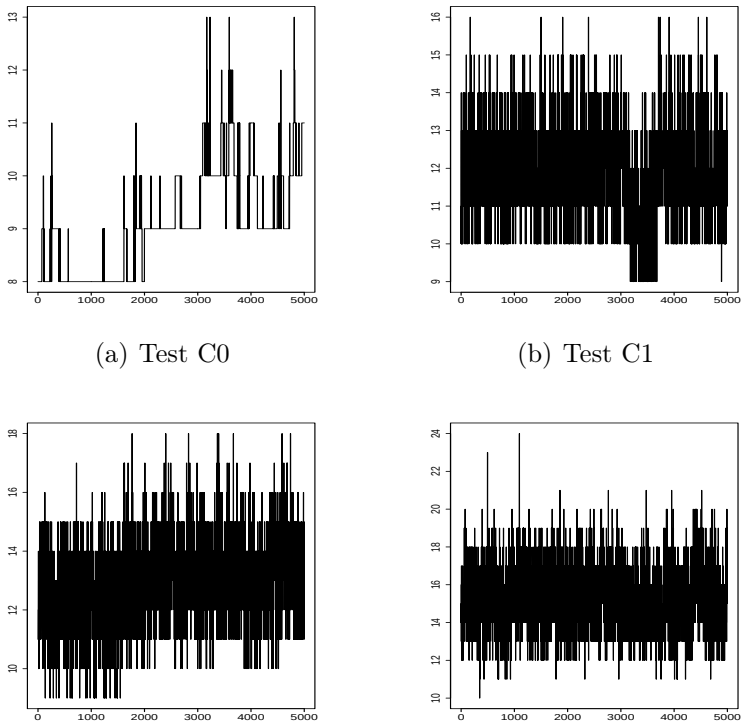


(d) Test C3: $\kappa = 10$

Figure 4.14: Histograms of the variable σ: the balancing effect of σ is clear observing the support of the posterior chains.

In Figure 4.15, some traceplots of K_n are shown: the mixing gets better increasing κ. Even if chains of N_ε are not reported for brevity, they are good; the number of components becomes very large when κ is big and the computational time goes up.

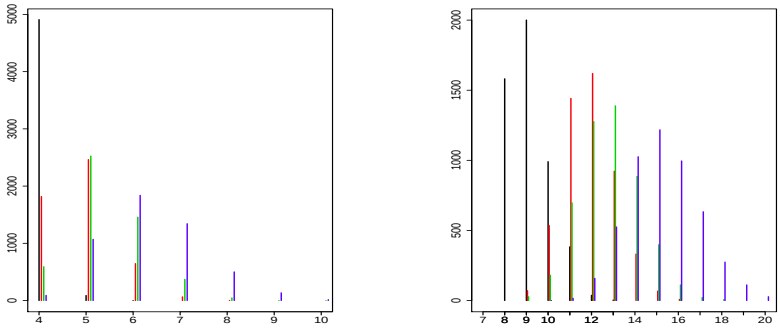
On the other hand, in tests d and D we put a non-informative prior on parameter κ: a Gamma distribution with shape 2 and rate 0.1. In these experiments, the mixing of the posterior chains gets worse drastically when σ assumes large values (as in d3 and D3): the correlation between variables is high and the chains get stuck into some states. Furthermore, the posterior



(a) Test C0 (b) Test C1

(c) Test C2 (d) Test C3

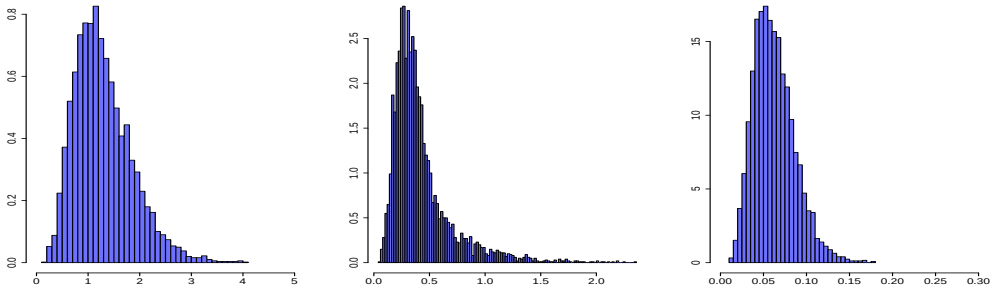
Figure 4.15: Traceplots of the variable K_n, number of groups, in tests C.



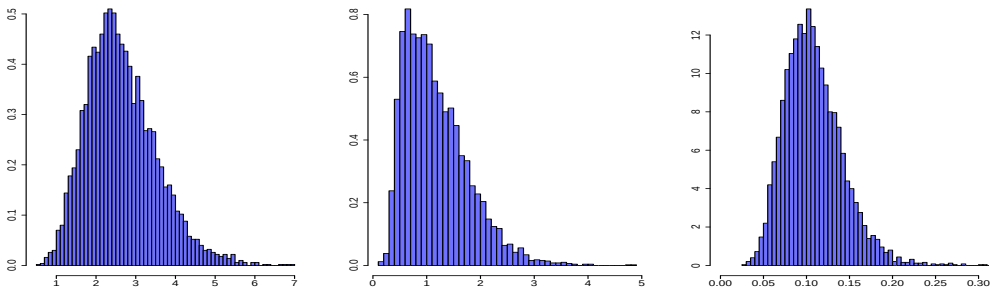
(a) Tests c (b) Tests C

Figure 4.16: Histograms of K_n are superimposed: the black ones correspond to tests c0 and C0 ($\kappa = 0.1$), the red ones to tests c1 and C1 ($\kappa = 1$). In green the tests c2 and C2 where κ is equal to 3, while the blue ones are c3 and C3 with $\kappa = 10$. It is clear the progressive shifting towards larger values.

mode of the variable K_n is always equal to 6 in tests d, equal to 12 in tests D.



(a) d0: $\sigma = 0.001$ (b) d2: $\sigma = 0.2$ (c) d3: $\sigma = 0.5$



(d) D0: $\sigma = 0.001$ (e) D2: $\sigma = 0.2$ (f) D3: $\sigma = 0.5$

Figure 4.17: Histograms of variable κ in some tests.

On the other hand, the chains of variable κ are satisfactory: Figure 4.17 shows the posterior distributions of this parameter. As usual, when σ is small (d0 and D0) κ assumes relatively large values, while if σ is large small values are assumed by the variable κ, proving the flexibility of the model.

Experiments e0 and E0 are interesting: here, both σ and κ are random. Priors are non-informative. Figure 4.18 shows the posterior behavior of the two variables in the case E0. Since κ has a prior with huge mean and variance, σ assumes small values, thus reducing the computational effort of the algorithm and favoring the good mixing of the chains. In fact, the posterior chain of K_n is good and the number of components of the mixture N_ε is neither too small nor too large, which is a good feature from a computational point of view (see Figure 4.19). We obtained similar results for test e0.

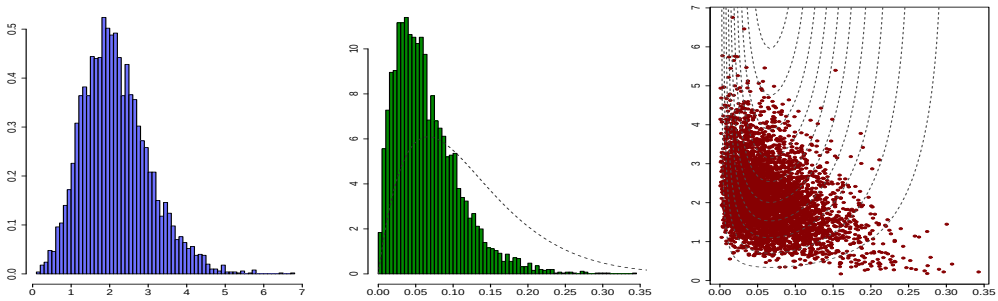


Figure 4.18: Histograms of variables κ (left) and σ (center). Scatterplot of the two variables, σ versus κ (right). In gray the priors.

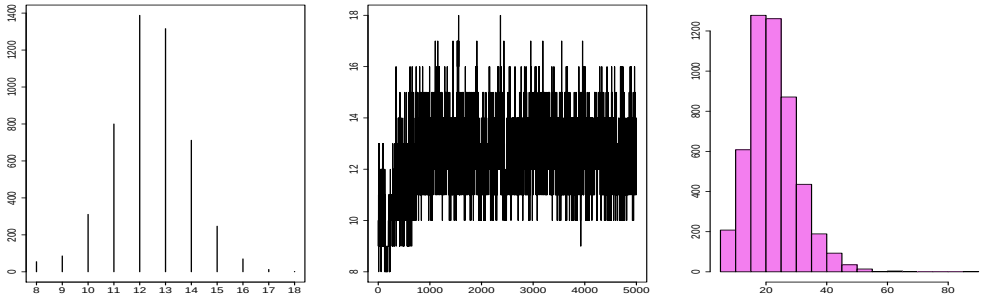


Figure 4.19: Histogram of the number of groups K_n (left) and its traceplot (center). On the right the histogram of the variable N_ε in test E0.

4.2 | A Bayesian nonparametric model-based clustering

We introduce here a simple method for cluster analysis which uses ε-NGG mixture models: it is a model-based technique, since it requires a mathematical model describing the problem.

In parametric model-based clustering, data are often modelled by a finite mixture of kernel densities, so that the number of clusters (i.e., the number of components in the mixture) is assumed fixed. In this sense a nonparametric approach allows more flexibility and robustness in the analysis.

We have already pointed out how ties in the sample $\boldsymbol{\theta}$ induce a partition π of the data, based on the values of latent variables θs. In particular, two points X_s and X_g belong to the same cluster if and only if $\theta_s = \theta_g$. In order to cluster data, is enough to know which latent variables are equal and which ones are not, avoiding the knowledge of their specific values.

The problem is to find one suitable posterior estimate of the partition of the data, the "best" posterior partition π. Usually, the standard choice is to minimize a loss-function. With this approach, an appropriate function is proposed, which evaluates the loss resulting from choosing the considered partition among all possibilities. The aim of this method is to find the clusterization of the data minimizing the posterior expected value of the loss-function. They are called loss-function minimization methods. See Cremaschi (2012) for a detailed review of the most used loss-functions in literature.

Often the loss function takes into account the misclassification costs generated by selecting a particular partition $\hat{\pi}$ instead of the true partition π. Recalling that $\boldsymbol{\theta}$ can be equivalently represented by the couple $(\boldsymbol{\phi}, \mathbf{c})$ where $\boldsymbol{\phi}$ are the unique values and $\mathbf{c}$ is the label vector that for a given partition contain the label associated with each observation, the loss function we are going to consider is the following:

$$L(\pi, \hat{\pi}) = \sum_{i<j} \left(a \mathbb{1}_{(c_i = c_j, \hat{c}_i \neq \hat{c}_j)} + b \mathbb{1}_{(c_i \neq c_j, \hat{c}_i = \hat{c}_j)} \right) \tag{4.1}$$

where $\hat{\pi}$ and $\hat{\boldsymbol{c}}$ stand for the estimated partition and label vectors and π and $\mathbf{c}$ for the "true" ones. From (4.1) we see that a and b are two parameters that weight two types of misclassification (or wrong labeling): a is the cost related to put in different groups elements that belong to the same group, while the weight b is related to the error of associate elements that belong to different groups.

Starting from the label vectors it is possible to build the corresponding incidence matrix: this is a matrix whose entries are binary values indicating whether two elements are in the same cluster or not. Integrating out the random part of $L(\pi, \hat{\pi})$, we obtain the posterior expected value of the loss-function:

$$l(\hat{\pi}) = \mathbb{E}\left(L(\pi, \hat{\pi})\right) = \sum_{i<j} \left(a\mathbb{1}_{(\hat{c}_i \neq \hat{c}_j)}\mathbb{P}(c_i = c_j|data) + b\mathbb{1}_{(\hat{c}_i = \hat{c}_j)}\mathbb{P}(c_i \neq c_j|data)\right).$$

Now l is a function of the proposed partition only and can be evaluated: the quantities $\mathbb{P}(c_i = c_j|data)$ and $\mathbb{P}(c_i \neq c_j|data)$ are estimated by the algorithm.

If the posterior coincidence probabilities ρ_{ij} are equal to $\mathbb{P}(c_i = c_j|data)$ and $\hat{K} := \frac{b}{a+b} \in [0, 1]$, the posterior expected value $l(\hat{\pi})$ can be written as

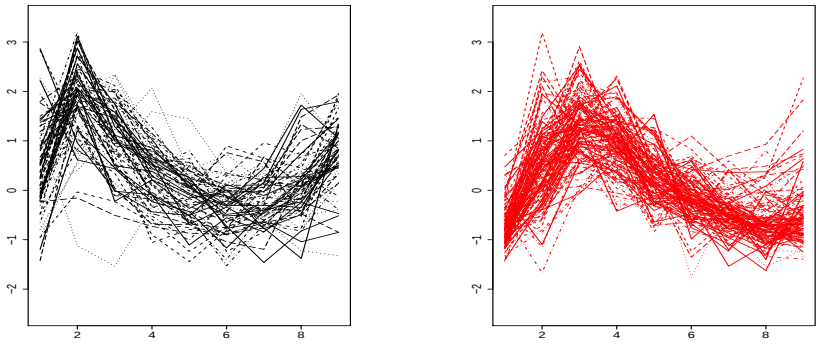
$$l(\hat{\pi}) = a\sum_{i<j} \rho_{ij} - (a+b)\sum_{i<j} \mathbb{1}_{(\hat{c}_i = \hat{c}_j)} \left(\rho_{ij} - \hat{K}\right), \tag{4.2}$$

therefore minimizing $l(\hat{\pi})$ corresponds to maximizing the function $f(\hat{\pi}) = \sum_{i<j} \mathbb{1}_{(\hat{c}_i = \hat{c}_j)} \left(\rho_{ij} - \hat{K}\right)$ with respect to $\hat{\pi}$. In the algorithm we need half of the iterations to estimate the quantity ρ_{ij} as a mean of incidence matrixes, and the other half to evaluate the function $f(\hat{\pi})$ and find the partition that maximize this value. In our examples, we set $\hat{K} = 0.5$, weighing the two kind of misclassification in the same way.

4.3 | Cluster analysis of the dataset

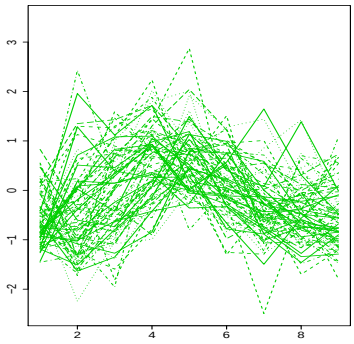
In this section we apply the method of Section 4.2 to the Yeast cell cycle data. These data are very used for clustering applications: we will use as reference partition that one of Cho et al. (1998) in Figure 4.20. They grouped the data by visual inspection according to the peak times of expression levels. In particular, they detected five peaking points in the second cycle, related to five phases of the cell cycle, at times 2, 3, 4, 6 and 8. In order to assess the clustering estimates we introduce here two cluster validation indexes well-known in literature: the Silhouette coefficient and the Adjusted Rand index.

Silhouette coefficient The Silhouette index assesses the quality of the proposed clusterization using only quantities relative to the dataset. It is a popular validation tool, first introduced by Rousseeuw in 1986. In particular, given a distance among the data (we will use euclidean distance) and a partition $\pi = \{C_1, \ldots, C_k\}$, the silhouette coefficient for an individual point

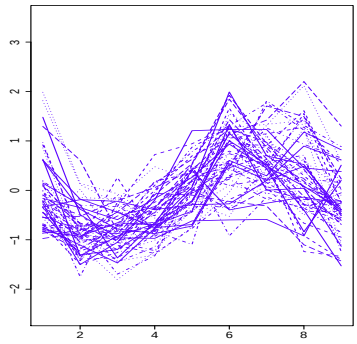


(a)

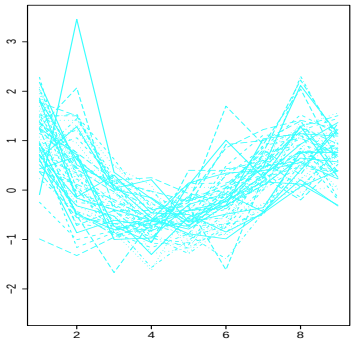
(b)



(c)



(d)



(e)

Figure 4.20: The partition of the data in 5 groups made by Cho et al. (1998) according to the time when the peak occurs.

can be computed as follows: first, for the $i-th$ datum, calculate the sample mean of the distance between the datum and all the others in its group. Call this value a_i. Secondly, compute the sample mean of the distances between the $i-th$ datum and all the points in a cluster not containing it. Find the minimum of that values with respect to all clusters: call this value b_i. Finally, the silhouette coefficient for the $i-th$ datum is defined as

$$s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}.$$

The value of the silhouette coefficient can vary between -1 and 1. Obviously, a large value reveals that the element is appropriately clustered. In fact, if $a_i = 0$, the silhouette coefficient of the $i-th$ observation is equal to 1. Moreover, a negative value is undesirable because corresponds to a case in which the mean distance of the element from the points in its cluster is greater than b_i , the minimum average distance to points in another cluster. An s_i near zero means that the datum is on the border of two clusters. An overall measure of the quality of a partition can be obtained by computing the average silhouette coefficient of all points: this value is reported for all the tests in the table. Note that, since the silhouette coefficient is not defined when there is a unique cluster, in this case we set it equal to 0. The average of the Silhouette index over all data of a cluster is a measure of how tightly grouped all the data in the cluster are. If there are too many or too few clusters, some of the clusters will typically display much narrower silhouettes than the rest. Thus silhouette plots and averages may be used to determine the natural number of clusters within a dataset.

Adjusted Rand Index The adjusted Rand index is a measure of the similarity between two data clusterings, one taken as reference. It is widely used in cluster validation analysis, when a "true" reference partition is available. Given a set of n elements and two partitions to compare, $\pi_1 = \{C_1, \ldots, C_k\}$ and $\pi_2 = \{B_1, \ldots, B_s\}$, consider the following quantities:

- a, the number of pairs of elements of the dataset that are in the same set both in π_1 and in π_2;
- b, the number of pairs of elements that are in two different sets both in π_1 and in π_2;
- c, the number of pairs of elements that are in the same set in π_1 but in different sets in π_2;
- d, the number of pairs of elements that are in different sets in π_1 but in the same set in π_2.

The Rand index (Rand, 1971) is defined as:

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$

where $a+b$ is the number of agreements between the two partitions, while $c+d$ is the number of times that the two disagree. Intuitively, R is the proportion of agreements between the two partition π_1 and π_2.

A form of the Rand index may be defined, that is adjusted for the chance grouping of elements: this is the adjusted Rand index. Suppose that the two partitions π_1 and π_2 to be compared are chosen randomly, with fixed number of groups and elements within each group. Hubert and Arabie (1985) defined the adjusted Rand index as

$$AR = \frac{R - \mathbb{E}(R)}{\max(R) - \mathbb{E}(R)}$$

which is bounded above 1 and takes the value 0 when the index equals its expected value. The two authors showed the following result under the assumption of a generalized hypergeometric model:

$$AR = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}}$$

where a_i and b_j are the numerosities of the groups in partitions π_1 and π_2 and n_{ij} are the agreements between the two clusters: in other words, it denotes the number of objects in common between C_i and B_j, i.e. $n_{ij} = \#\{C_i \cap B_j\}$. We will consider as reference partition that one of Cho et al. (1998) in Figure 4.20.

Let us see the cluster analysis provided by the model under some of the test experiments. Table 4.3 shows the estimated number of clusters as those minimizing $l(\hat{\pi})$ in (4.2), and the Silhouette and Adjusted Rand (AR) indexes from the posterior distribution of the partition π.

As we already pointed out in Section 4.1, the tests named with the capital letter have a P_0 (that strongly influences the number of groups) inducing a higher number of clusters than the tests named with the lower case letter. For this reason the values of the AR indexes, that assess the distance between the our clusterization and the one of Cho et al. (1998) (that identifies only 5 clusters), is lower for all the tests with the capital letter. However, we recall that the clusters provided by Cho et al. (1998) are not the "true" ones, but the result of a clustering made by visual inspection.

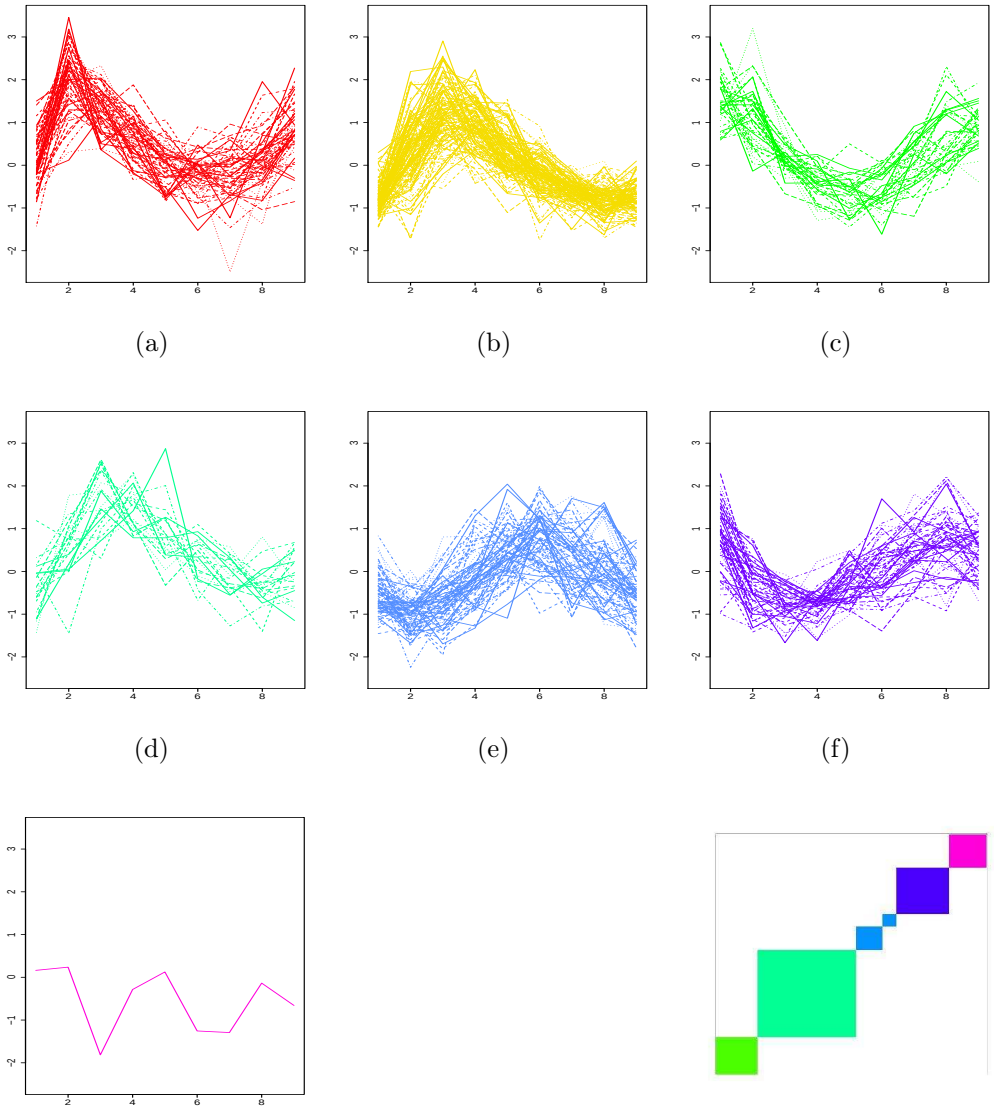


Figure 4.21: Data clustering using test a0: 7 clusters are found by our loss function minimization method. Image (h) represents the incidence matrix.

Name	Nr. of clusters	Silhouette Index	Adj Rand Index
a0	7	0.199	0.444
A0	11	0.121	0.376
a2	8	0.19	0.445
A2	11	0.108	0.38
b0	6	0.2	0.44
B0	10	0.133	0.372
c2	9	0.183	0.446
C2	15	0.091	0.369
d0	7	0.195	0.443
D0	11	0.099	0.369
e0	6	0.196	0.443
E0	13	0.096	0.37
f3	6	0.199	0.443
F3	10	0.126	0.373
g1	7	0.2	0.443
G1	10	0.129	0.373

Table 4.3: Table of the tests for which the clustering algorithm has been applied: for every choice of the parameters the number of clusters and the value of the two validation indexes are reported.

Also the Silhouette index is lower, in general, for the tests named with capital letter: we will see later through some examples that the estimates seem better in the tests with the lower case letter. In fact, in that cases, too groups are identified by the algorithm and some of them could be clustered together, thus leading to a small Silhouette coefficient.

In Figure 4.21 the posterior cluster estimates for test a0, where all the parameters are fixed, are shown. The groups identified by Cho et al. (1998) (compare Figure 4.20) are similar to these: the only cluster that it is not really distinguishable is group (c) of Cho, since presents a large empirical variance and it does not show a precise feature. An interesting characteristic is the splitting of group (b) of Cho (Fig. 4.20) in two clusters by our model: in fact, our model takes into account the general behavior of the curves and not only where the peak occurs, differently from the visual clusterization made by Cho et al. (1998). For this reason we got the splitting into two groups: both have a peak at time $t = 3$ but group (b) in Figure 4.21 behaves differently from the group (d) in the same figure with respect to the values of the curve at times 8 and 9.

Furthermore, this method identifies a cluster that it is not present in the work of Cho et al. (1998): cluster (c) which presents two maximums at time $t = 1$ and $t = 9$. Group (g) is formed by a unique curve: the algorithm

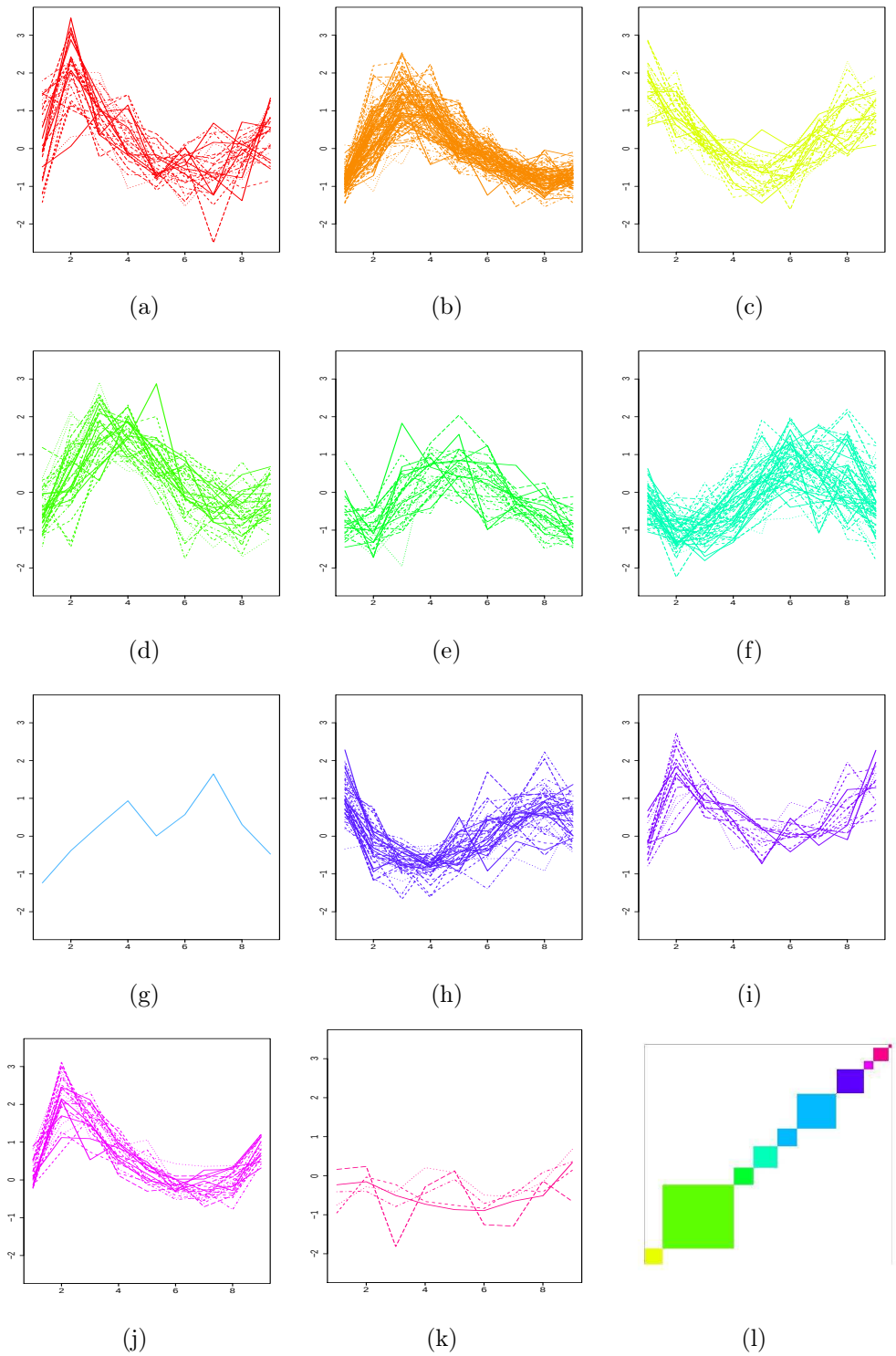


Figure 4.22: Data clustering in test A0: 11 clusters are found by the loss function minimization method. Image (l) represents the incidence matrix.

seems to suggest that this is a sort of outlier of the dataset, since it does not resemble any other datum because it has no peaks. The same feature occurs in other tests, for instance a2, d0 and g1.

Tests with the capital letters identify more clusters in the dataset, with a smaller variance inside the group: for example, the inference of test A0 is presented in Figure 4.22. The groups identified by Cho et al. (1998) are divided into many clusters: for example, cluster (a) of Cho (Fig. 4.20) seems here to be split into three groups (Fig. 4.22: (a), (i), (j)). All have a peak in $t = 2$ but the behavior of the curves in the other times changes. However, this model identifies too many clusters: in fact, probably groups (i) and (j) could be clustered together, since they do not present many differences.
We report here the Silhouette indexes for each group in Figure 4.22:

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)
-0.01	0.18	0.21	0.2	0	0.09	0.04	-0.02	0.14	0.1	0.2

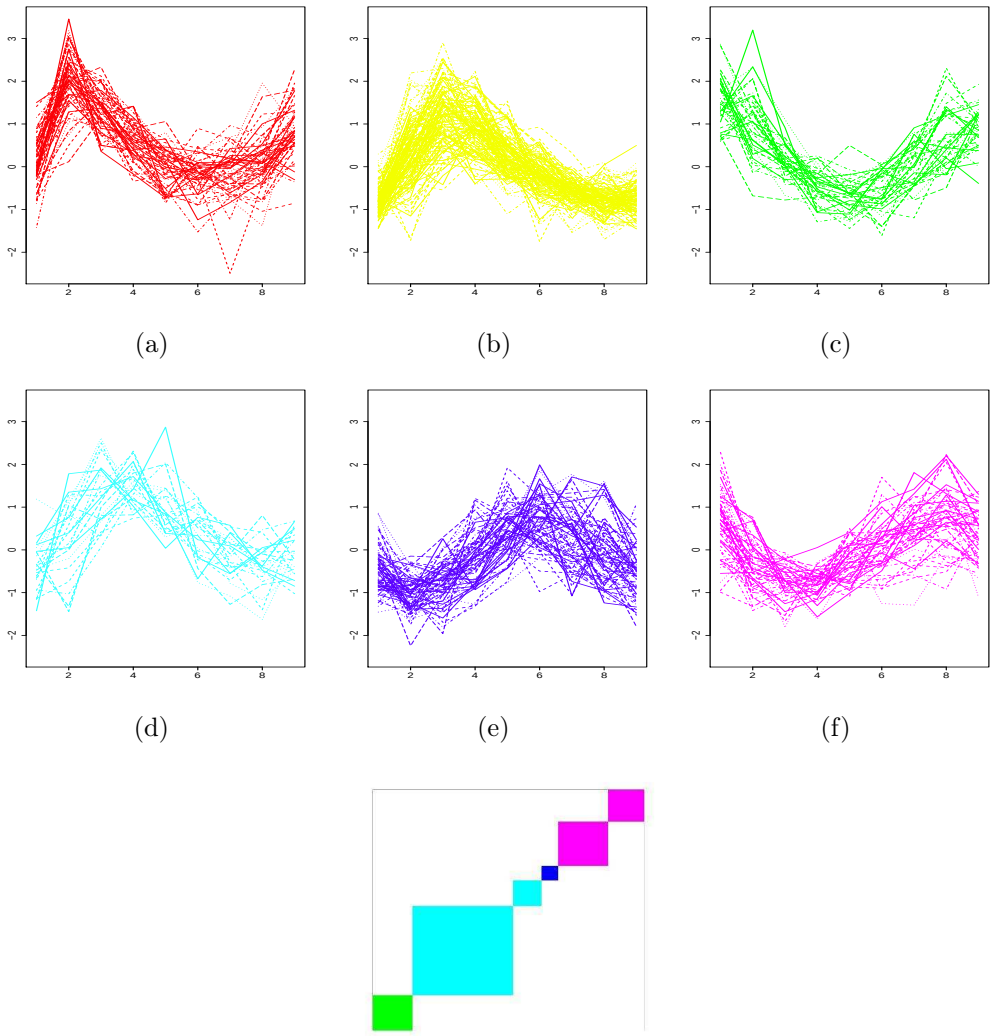
The Silhouette index of the first cluster is negative: this means, according to this coefficient, that the elements of the group are less distant from other clusters, thus they could be clustered with other groups.

Clustering estimates arising from test e0 are satisfactory: the Silhouette coefficient is relatively high in every group, except from (d) which is the cluster presenting the larger variability,

(a)	(b)	(c)	(d)	(e)	(f)
0.22	0.23	0.22	0.04	0.18	0.14

thus indicating a good estimation. This time no "outliers" are detected and all the clusters have a good numerosity. The first two clusters are very similar to the ones of Cho et al. (1998). On the other hand, test E0 identifies too many clusters (see Figure 4.24): σ and κ here were random variables, conducing to a lot of clusters with a small numerosity.

In conclusion, we have seen how the method seems robust with respect to the choice of the parameters ε, σ and κ (fixed or random). What really influences is the choice of P_0: this is a problem that affects in general the non parametric mixture models. In particular, the choice becomes more difficult when the dimension p of the problem goes up, since slight differences in the choice of the distribution P_0 lead to enormous changes in the results.



(a) (b) (c)

(d) (e) (f)

(g)

Figure 4.23: Data clustering estimates for test e0: 6 clusters are found by the loss function minimization method. Image (g) represents the incidence matrix.

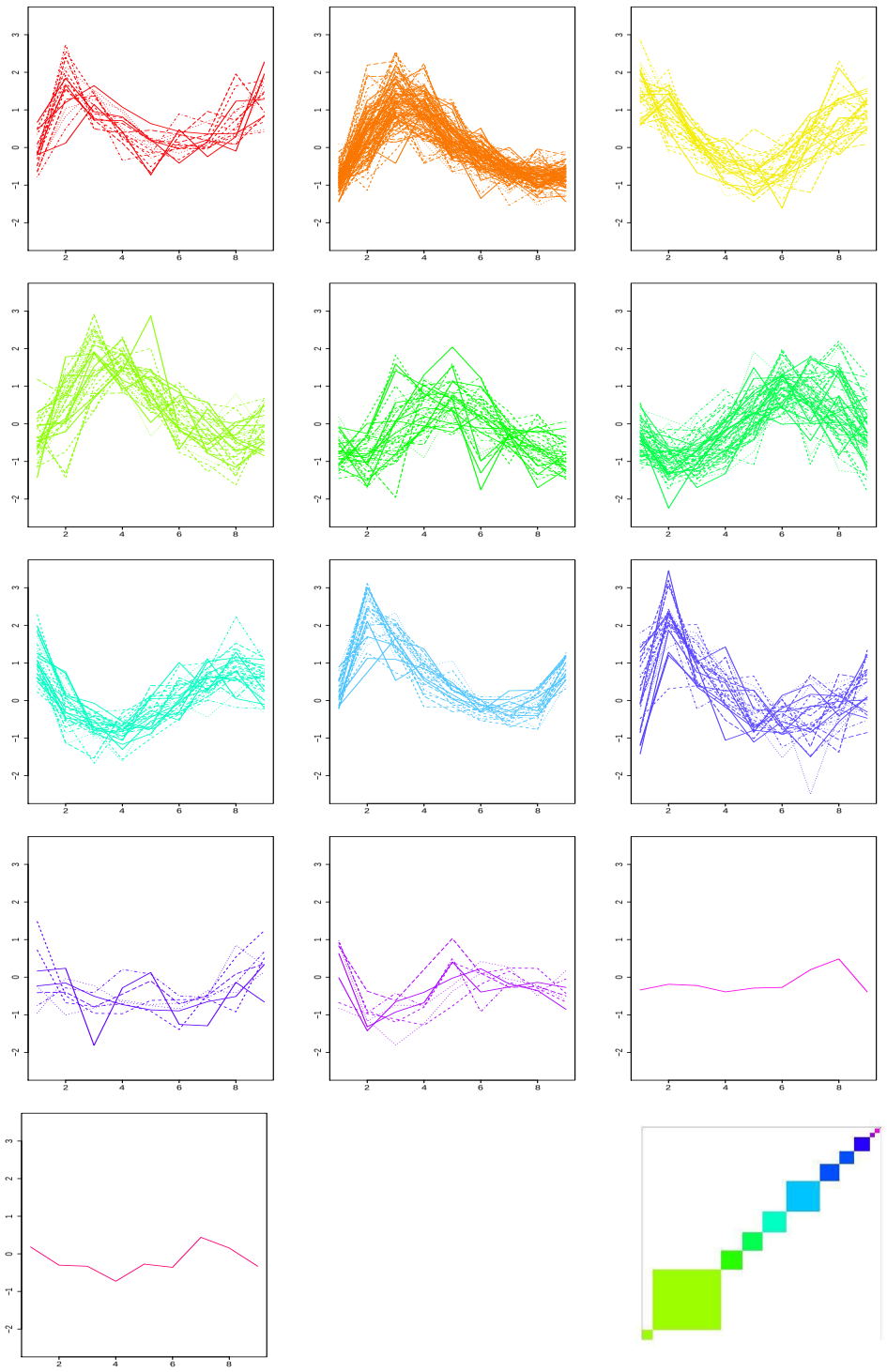


Figure 4.24: Data clustering estimates for test E0. The last image represents the incidence matrix.

Conclusions and future developments

This work has proposed a new model for density and cluster estimates in the Bayesian nonparametric framework. In particular, we introduced a finite dimensional process called ε-NGG process that, when ε tends to 0, converges in distribution to the well-known Normalized Generalized Gamma process.

We employed this process as the mixing measure in a mixture model for density estimation. An interesting achievement is that, varying ε, a large range of models can be obtained: from a nonparametric NGG mixture model, when ε decreases to 0, to a parametric model, where the number of elements of the mixture is fixed, when ε assumes large values. Hence, on the one hand, the model can be used as an approximation of a NGG mixture model on which many theoretical results are available in the literature. For instance, the distribution of the number of distinct values in a sample from the process is known: this turns out to be useful when the parameters of the prior must be fixed. On the other hand, the model can be viewed as a separate model with a new prior: since it is finite dimensional, the inference will be relatively simple. Furthermore, the precision parameter ε can be considered as a random variable, once we have elicited a prior for it: in this case, the data "drive" the degree of approximation. Of course under this model the posterior distribution must be computed via simulation methods: a Gibbs sampler algorithm has been built to this aim. All the updating steps are relatively easy to implement and the model is more flexible than the popular DPM model. In addition, thanks to the finite approximation, there is no need to integrate out from the model the mixing component (i.e. the infinite dimensional parameter), thus pursuing a *full nonparametric Bayesian inference*, obtaining posterior estimates of linear and non linear functionals of the population distribution.

We illustrated our proposal through a density estimation problem: thanks to a deep robustness analysis, the role and the influence of the parameters ε, σ and κ of our prior on the posterior chains and estimates have been

understood; moreover, the robustness of the model with respect to the choice of the hyperparameters has been verified.

In addition to density estimation, a clustering problem has also been tackled in the multivariate case: in fact, the ε-NGG mixture model can be useful to solve clustering problems, minimizing a-posteriori a suitable loss function (of the random partition), that quantifies the loss when a misclassification error occurs. The obtained cluster estimates were satisfactory.

As far as the drawbacks of the model are concerned, the first issue consists in the choice of the distribution P_0, that is a parameter of the ε-NGG process. Particularly when the dimension of the data is high, the choice of this distribution is very difficult and affects both the estimates and the mixing of the MCMC chains. However, this is a problem troubling nonparametric mixture models in general. A second problem concerns the parameter σ: when it assumes values close to 1, the computation becomes difficult because of the presence of the Incomplete Gamma functions in the algorithm, which are very unstable in this case. Moreover, the number of components in the mixture grows very fast with σ, slowing the run-time of the algorithm. Finally, another problem is the slow convergence and the bad mixing of posterior chains (especially in the multivariate case), requiring a long burn-in period and a large thinning in the algorithm.

As future developments, the parallelization of the C++ code could be interesting to speed up the algorithm, which is fast in the unidimensional case, but it can be very slow in the multivariate case because of the presence of sampling from multivariate distributions.

Furthermore, different loss functions or even different types of cluster estimates could be used in the clustering problem, in order to improve the estimation when too clusters are selected by the model.

Bibliography

Argiento, R., Guglielmi, A., and Pievatolo, A. (2010). Bayesian density estimation and model selection using nonparametric hierarchical mixtures. *Comput. Statist. Data Anal.*, 54(4):816–832.

Barrios, E., Lijoi, A., Nieto-Barajas, L. E., and Pruenster, I. (2012). Modeling with normalized random measure mixture models. *Carlo Alberto Notebooks*, (276).

Cho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gabrielian, A., Landsman, D., Lockhart, D., and Davis, R. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*.

Cremaschi, A. (2012). Model-based clustering via bayesian nonparametric mixture models. *Tesi di laurea magistrale, Ingegneria Matematica, Politecnico di Milano*.

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *J. Amer. Statist. Assoc.*, 90(430):577–588.

Favaro, S. and Teh, Y. W. (2013). MCMC for normalized random measure mixture models. *Statistical Science*, 28(3):335–359.

Gelfand, A. E. and Kottas, A. (2002). A computational approach for full nonparametric Bayesian inference under Dirichlet process mixture models. *J. Comput. Graph. Statist.*, 11(2):289–305.

Griffin, J. E. (2013). An adaptive truncation method for inference in bayesian nonparametric models. *arXiv preprint arXiv:1308.2045*.

Griffin, J.E. and Walker, S. G. (2011). Posterior Simulation of Normalized Random Measure Mixtures. *Journal of Computational and Graphical Statistics*.

Ishwaran, H. and James, L. F. (2001). Gibbs sampling methods for stick-breaking priors. *J. Amer. Statist. Assoc.*, 96(453):161–173.

Ishwaran, H. and Zarepour, M. (2000). Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models. *Biometrika*.

Ishwaran, H. and Zarepour, M. (2003). Random probability measures via Polya sequences: revisiting the Blackwell-MacQueen urn scheme. *arXiv preprint math/0309041*.

James, L. F., Lijoi, A., and Prünster, I. (2009). Posterior analysis for normalized random measures with independent increments. *Scand. J. Stat.*, 36(1):76–97.

Kingman, J. F. C. (1993). Poisson processes, volume 3 of *Oxford Studies in Probability*. The Clarendon Press Oxford University Press, New York. Oxford Science Publications.

Lijoi, A., Mena, R. H., and Prünster, I. (2007). Controlling the reinforcement in Bayesian non-parametric mixture models. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 69(4):715–740.

MacEachern, S. N. (1998). Computational methods for mixture of Dirichlet process models. In *Practical nonparametric and semiparametric Bayesian statistics*, volume 133 of *Lecture Notes in Statist.*, pages 23–43. Springer, New York.

Muliere, P. and Tardella, L. (1998). Approximating distributions of random functionals of Ferguson-Dirichlet priors. *Canad. J. Statist.*, 26(2):283–297.

Müller, P. and Mitra, R. (2013). Bayesian nonparametric inference—why and how. *Bayesian Anal.*, 8(2):269–302.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *J. Comput. Graph. Statist.*, 9(2):249–265.

Papaspiliopoulos, O. and Roberts, G. O. (2008). Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika*, 95(1):169–186.

Pitman, J. (1996). Some developments of the Blackwell-MacQueen urn scheme. In *Statistics, probability and game theory*, volume 30 of *IMS Lecture Notes Monogr. Ser.*, pages 245–267. Inst. Math. Statist., Hayward, CA.

Pitman, J. (2003). Poisson-Kingman partitions. In *Statistics and science: a Festschrift for Terry Speed*, volume 40 of *IMS Lecture Notes Monogr. Ser.*, pages 1–34. Inst. Math. Statist., Beachwood, OH.

Pitman, J. (2006). Combinatorial stochastic processes, volume 1875 of *Lecture Notes in Mathematics. Springer-Verlag, Berlin.* Lectures from the 32nd Summer School on Probability Theory held in Saint-Flour, July 7–24, 2002, With a foreword by Jean Picard.

Pitman, J. and Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Ann. Probab.*, 25(2):855–900.

Regazzini, E., Lijoi, A., and Prünster, I. (2003). Distributional results for means of normalized random measures with independent increments. *Ann. Statist.*, 31(2):560–585. Dedicated to the memory of Herbert E. Robbins.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statist. Sinica*, 4(2):639–650.

Walker, S. G. (2007). Sampling the Dirichlet mixture model with slices. *Comm. Statist. Simulation Comput.*, 36(1-3):45–54.

Wolpert, R. L. and Ickstadt, K. (1998). Simulation of Lévy random fields. In *Practical nonparametric and semiparametric Bayesian statistics*, volume 133 of *Lecture Notes in Statist.*, pages 227–242. Springer, New York.