Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

DATA SCIENCE AND COMPUTATION

Ciclo 34

Settore Concorsuale: 13/D1 - STATISTICA

Settore Scientifico Disciplinare: SECS-S/01 - STATISTICA

STATISTICAL LEARNING OF RANDOM PROBABILITY MEASURES

Presentata da: Mario Beraha

Coordinatore Dottorato	**Supervisore**
Daniele Bonacorsi	Alessandra Guglielmi

Esame finale anno 2023

Acknowledgements

Research is a journey, and I could not have asked for better adventure companions for this trip. First, I want to thank my supervisor and scientific mentor Alessandra Guglielmi, who believed in me and encouraged me in the first place to walk down this path, while continuously supporting me and fostering my curiosity, even when it meant leaving Bayesian nonparametrics (only for a short while!). It is not an exaggeration to say that meeting you changed my life. I also had the change to collaborate closely with other more experienced researchers: Raffaele Argiento, Federico Camerlenghi, Jim Griffin, Jesper Møller, and Fernando Quintana. Working with each of you has thought me valuable lessons. Most of all, I learned that doing research is fun even when you are a professor and that the excitement never goes away. I hope to feel the same in 20 years. Among my fellow PhD students, a special thanks goes to my friend Matteo Pegoraro: the days in the "aula lavagne" working on our Wasserstein ideas have been fun, and thought me the power of interdisciplinary research, we should do it more often! Thanks also to Riccardo Peli, Riccardo Corradin, Matteo Gianella, Lorenzo Ghilotti, and Bruno Guindani, research is always fun, but it's funnier when done with you. Thanks to my office mates and all the people on the second floor of the Nave, spending four years there would have been much duller without you.

Thanks to my lifetime friends Carlo, Lino, and Sandro, and to all other friends who shared drinks, nights out, and holidays with me: there are simply too many of you to acknowledge, but you know who you are. Finally, thanks to my parents, my sister Marina and her husband Lorenzo, and, especially, to my wife Francesca: I am grateful for having you by my side as my best friend, your support has been fundamental throughout these years. This thesis is dedicated you.

Abstract

The study of random probability measures is a lively research topic that has attracted interest from different fields in recent years. In this thesis, we consider random probability measures in the context of Bayesian nonparametrics, where the law of a random probability measure is used as prior distribution, and in the context of distributional data analysis, where, in the simplest setting, the goal is to perform inference given an independent and identically distributed sample from the law of a random probability measure.

The contributions contained in this thesis can be subdivided according to three different topics: (i) the use of almost surely discrete repulsive random measures (i.e., whose support points are well separated) for Bayesian model based clustering, (ii) the proposal of new laws for a collection of random probability measures to be used for Bayesian density estimation in the context of partially exchangeable data subdivided into different groups, and (iii) the study of principal component analysis and regression models for probability distributions seen as elements of the 2-Wasserstein space. Specifically, for point (i) above we propose an efficient Markov chain Monte Carlo algorithm for posterior inference, which sidesteps the need of split-merge reversible jump moves typically associated with poor performance, we propose a model for clustering high-dimensional data by introducing a novel class of anisotropic determinantal point processes, and study the distributional properties of the repulsive measures, shedding light on important theoretical results which enable more principled prior elicitation and more efficient posterior simulation algorithms. For point (ii) above, we consider several models suitable for clustering homogeneous populations, inducing spatial dependence across groups of data, extracting the characteristic traits common to all the data-groups, and propose a novel vector autoregressive model tailored to the study of growth curves of Singaporean kids. Finally, for point (iii), we propose a novel class of "projected" statistical methods for distributional data analysis for measures on the real line and on the unit-circle respectively.

Contents

Summary

This thesis is concerned with the study of random probability measures (RPMs) from a statistical point of view. To give a concise definition, consider a complete and separable metric space $(\mathbb{X}, d)$; as the name suggests, an RPM is a random variable taking values on $(\mathbb{P}_{\mathbb{X}}, \mathcal{P}_{\mathbb{X}})$, that is, the space of probability measures over $\mathbb{X}$, endowed with its Borel σ-field $\mathcal{P}_{\mathbb{X}}$.

RPMs appear in several fields, such as Bayesian nonparametrics, distributional data analysis, and stochastic geometry (Keeler and Błaszczyszyn, 2014; Caron and Fox, 2017). For example, in Bayesian nonparametrics (Müller et al., 2015), under the assumption of exchangeability, it is common to consider models for data X_i, $i = 1, \ldots, n$ like the following:

$$\begin{aligned} X_1, \ldots X_n \mid \tilde{p} &\overset{\text{iid}}{\sim} \tilde{p} \\ \tilde{p} &\sim Q \end{aligned} \tag{1}$$

where $\tilde{p}$ represents the population distribution of the data. See Chapter 1 for further details and justification. We can think of the random measure $\tilde{p}$ in (1) as an "infinite-dimensional" parameter, for which, under the Bayesian paradigm, a prior must be selected. Therefore, the measure Q over $(\mathbb{P}_{\mathbb{X}}, \mathcal{P}_{\mathbb{X}})$ (that is, is the law of a random probability measure) acts as the prior distribution. In more complex settings, extensions of model (1) have been considered to account for the dependence on covariates. For example, when each datum is associated with a categorical covariate acting as a group indicator, we can divide the observations into those groups and write the sample as $(X_{j,i},\ j = 1, \ldots, g,\ i = 1, \ldots, n_j)$. Under the assumption of partial exchangeability, a nonparametric Bayesian model is:

$$\begin{aligned} X_{j,1}, \ldots X_{j,n_j} \mid \tilde{p}_j &\overset{\text{iid}}{\sim} \tilde{p}_j, \qquad j = 1, \ldots, g \\ (\tilde{p}_1, \ldots, \tilde{p}_g) &\sim Q \end{aligned} \tag{2}$$

where Q is a probability measure over the product space $(\mathbb{P}_{\mathbb{X}})^g$. See Chapter 5 for details on partial exchangeability and several examples of the measure Q.

In the Bayesian setting, "statistical learning" is obtained by computing the posterior distribution of $\tilde{p}$ (or of the vector $(\tilde{p}_1, \ldots, \tilde{p}_g)$), that is, the conditional of $\tilde{p}$ law given observations. Usually, it is impossible to derive an analytical expression in closed form for such a posterior distribution, so that an approximation must be constructed, for instance, by means of Markov chain Monte Carlo simulation.

Distributional data analysis is a recent field that is concerned with extracting information from a set of probability distributions. In distributional data analysis, observations themselves are probability measures. Considering the observations as random variables makes them RPMs. Several challenges must be faced when approaching this problem: for example, the space of distributions $\mathbb{P}_{\mathbb{X}}$ is not linear. Hence, classical statistical techniques cannot be adapted to the analysis of distributional data.

This thesis is divided into three main parts and a total of 13 self-contained chapters. Its content is summarized below. The first part considers models such as (1) and focuses on the specification of nonparametric priors in connection with Bayesian model-based clustering.

Chapter 1 provides an introduction to Bayesian nonparametrics in the exchangeable setting and sets the stage for the study of “repulsive” measures Q.

Chapter 2 sets a general framework for repulsive mixture models and proposes a novel MCMC algorithm for posterior inference. In particular, our algorithm avoids complex split-merge birth-death reversible jumps MCMC moves, making it easier to extend it to models we have not considered there. Furthermore, we demonstrate superior performance compared to alternative algorithms and argue for the usefulness of a “repulsive” measure Q in misspecified mixture models. This chapter is based on a recently published paper; for more details, see Mario Beraha, Raffaele Argiento, Jesper Møller, and Alessandra Guglielmi (2022). “MCMC computations for Bayesian mixture models using repulsive point processes.” In: *Journal of Computational and Graphical Statistics*.

Chapter 3 studies the distributional properties of repulsive measures and their use in Bayesian mixtures. It sheds light on important theoretical results that characterize the prior and posterior of $\tilde{p}$. These results can be used for prior elicitation and to derive new MCMC algorithms.

Chapter 4 extends Chapter 2 to the setting of clustering of high-dimensional data. Inference for high-dimensional data is notoriously complex because of the large number of parameters involved. We propose a latent factor model similar to Chandra et al. (2020), where instead of a Dirichlet process mixture, we propose a repulsive mixture to cluster the latent factors. The main contribution is the definition of an anisotropic determinantal point process, which, used in combination with the latent factor model, leads to well-separated clusters of data.

In the second part of the thesis, we consider models that induce dependence on a collection of RPMs through covariates.

Chapter 5 consists of an introduction to dependent random probability measures.

Chapter 6 proposes a prior Q as in (2) for a collection of RPMs such that, with positive probability $\tilde{p}_i = \tilde{p}_j$, while avoiding the degeneracy issue of nested processes discussed in Camerlenghi et al. (2019). We propose an efficient MCMC algorithm and apply our model to detect homogeneous groups of data. Several theoretical properties of the model are investigated. This chapter is based on the published article: Mario Beraha, Alessandra Guglielmi, and Fernando A. Quintana (2021). “The semi-hierarchical Dirichlet Process and its application to clustering homogeneous distributions”. In: *Bayesian Analysis*

Chapter 7 considers the case in which each group of data is associated with a spatial location. It proposes a prior Q that favors similar distributions in nearby locations. The model is applied to the analysis of Airbnb properties in the city of Amsterdam. A published version of the paper can be found in Mario Beraha, Matteo Pegoraro, Riccardo Peli, and Alessandra Guglielmi (2021). “Spatially dependent mixture models via the logistic multivariate CAR prior”. In *Spatial Statistics*.

Chapter 8 deals again with grouped data but considers specifically the setting where the number of groups g is large (possibly much larger than the number of observations in each group). Instead of proposing a very flexible model, we focus on obtaining interpretable posterior summaries that can be used to explore and explain the difference in distributions across different groups, considering a “latent factor” model for a collection of random probability measures. This chapter is based on the preprint Mario

Beraha and Jim E. Griffin (2022). “Normalized latent measure factor models.” In: *arXiv:2205.15654*.

Chapter 9 presents an application to clustering growth curves of kids in Singapore. To this end, we propose a vector autoregressive model in which the patient-specific autoregression matrix is given a logit stick-breaking prior. This allows us to group patients according to their autoregression matrix while inducing dependence on subject-specific covariates. See the preprint Mario Beraha, Alessandra Guglielmi, Fernando A. Quintana, Maria de Iorio, Johan Eriksson, and Fabian Yap (2022). “'Bayesian nonparametric vector autoregressive models via a logit stickbreaking prior: an application to child obesity”. In: *arXiv:2203.12280*.

The third part of the thesis deals with distributional data analysis.

Chapter 10 gives an introduction to the analysis of distributions, focusing in particular on the challenges faced by the nature of $\mathbb{P}_{\mathbb{X}}$.

Chapter 11 considers distributions on $\mathbb{R}$ and proposes a *projected* framework for PCA and linear regression when considering distributions in the Wasserstein space. Our approach exploits the particular structure of the Wasserstein space of one-dimensional distribution to derive a fast implementation. This chapter is based on the paper Matteo Pegoraro and Mario Beraha (2022). “Projected Statistical Methods for Distributional Data on the Real Line with the Wasserstein Metric”. In: *Journal of Machine Learning Research.*

Chapter 12 is an extension of the previous one to the simplest “nontrivial” setting outside $\mathbb{R}$, that is, the circumference. After establishing some new results for the optimal transport maps for measures on the circumference, we propose a framework for PCA, motivated by the study of a dataset of the optical nerve width.

Finally, Chapter 13 presents `BayesMix`: a `C++` library which implements efficient and extensible algorithms for posterior inference in Bayesian mixture models. See the preprint Beraha Mario, Bruno Guindani, Matteo Gianella, and Alessandra Guglielmi (2022). “`BayesMix`: Bayesian Mixture Models in `C++`”. In *arXiv:2205.08144*.

1. Beyond CRMs: normalized random measures with atoms' interaction for Bayesian mixture models

The first part of this thesis is dedicated to the study of exchangeable data, with the main objective being to obtain cluster estimates of data points in a model-based Bayesian nonparametric framework. This chapter gives a broad and informal overview of Bayesian nonparametrics. Then, we will argue for the use of *repulsive* normalized random measures, that is, discrete random probability measures with well-separated support points, as priors for Bayesian models for clustering.

Three contributions will be presented in the following chapters. Chapter 2, based on Beraha et al. (2022), joint work with Raffaele Argiento, Jesper Møller and Alessandra Guglielmi, presents a Markov chain Monte Carlo algorithm for posterior inference. Chapter 3, based on a joint work with Raffaele Argiento, Federico Camerlenghi, and Alessandra Guglielmi, discusses results regarding the distributional theory for normalized random measures based on marked point processes in Bayesian models. Finally, in Chapter 4, based on a joint work with Lorenzo Ghilotti and Alessandra Guglielmi, an extension of repulsive mixture models to high-dimensional settings by means of anisotropic determinantal point processes is presented.

1.1 Background

Exchangeability is an assumption on the data-generating process, stating that the order in which observations are recorded is not relevant, i.e. the distribution function of the joint law of the sample is symmetric in its arguments. Formally, we say that a sequence $X_1, X_2, \ldots$ (finite or infinite) of random variables is exchangeable if for any finite permutation σ of the indices, we have the following-

$$\mathcal{L}(X_1, X_2, \ldots) = \mathcal{L}(X_{\sigma(1)}, X_{\sigma(2)}, \ldots),$$

where we use $\mathcal{L}$ to generically represent the law of a (sequence of) random variable(s) and assume that the X_i's take value in a polish space.

When the sequence $X_1, X_2, \ldots$ is infinite, then, by de Finetti's theorem (de Finetti, 1938), exchangeability is equivalent to assuming the existence of a probability distribution Q such that the joint density of $X_1, \ldots, X_n$ equals

$$\mathsf{P}(X_1 \in \mathrm{d}x_1, \ldots, X_n \in \mathrm{d}x_n) = \int \prod_{i=1}^{n} \mathsf{P}(X_i \in \mathrm{d}x_i \,|\, \nu) Q(\mathrm{d}\nu). \tag{1.1}$$

The de Finetti representation theorem can be regarded as one of the main motivations for the Bayesian approach since it ensures the existence of a *likelihood* $\mathsf{P}(X_i \in \mathrm{d}x_i \,|\, \nu)$ and a *prior* distribution Q. In particular, we can advocate the use of nonparametric models following (1.1) where $\nu \sim Q$ is a *random probability measure* and $\mathsf{P}(X_i \in \mathrm{d}x_i \,|\, \nu) = \nu(\mathrm{d}x_i)$.

1.1.1 The Dirichlet Process

The most notable example of nonparametric prior distribution Q is the celebrated Dirichlet process (DP, Ferguson, 1973). To define the Dirichlet process, let $\alpha > 0$ and G_0 be a probability distribution over a complete and separable metric space $(\mathbb{X}, d)$ endowed with the usual Borel sigma algebra. Then we say that a random probability measure $\tilde{p}$ is distributed as the Dirichlet process with total mass parameter α and base measure G_0, if, for any measurable partition $\{A_1, \dots, A_n\}$ of $\mathbb{X}$ we have:

$$(\tilde{p}(A_1), \dots, \tilde{p}(A_n)) \sim \text{Dirichlet}_n \left(\alpha G_0(A_1), \dots, \alpha G_0(A_n)\right), \tag{1.2}$$

where Dirichlet_n denotes the Dirichlet distribution on the $n-1$ dimensional simplex. If (1.2) holds, we write $\tilde{p} \sim DP(\alpha, G_0)$, see Ferguson (1973).

From the finite-dimensional characterization (1.2), it might not be clear how to use $\tilde{p} \sim DP(\alpha, G_0)$ in a Bayesian model. The stick-breaking representation in Sethuraman (1994) provides a more intuitive characterization. In fact, $\tilde{p} \sim DP(\alpha, G_0)$ if and only if

$$\begin{gathered} \tilde{p}(\cdot) \overset{\mathrm{d}}{=} \sum_{h \geq 1} w_h \delta_{\tau_h} \\ \tau_1, \tau_2, \dots \overset{\text{iid}}{\sim} G_0, \qquad w_1, w_2, \dots \sim \text{SB}(\alpha). \end{gathered} \tag{1.3}$$

Here, $\text{SB}(\alpha)$ denotes the stick-breaking or GEM distribution, that is,

$$\begin{gathered} w_1 = \nu_1, \qquad w_j = \nu_j \prod_{\ell < j} (1 - \nu_\ell), \quad j = 2, 3, \dots \\ \nu_1, \nu_2, \dots \overset{\text{iid}}{\sim} \text{Beta}(1, \alpha) \end{gathered}$$

Yet another characterization of the DP is through the marginal distribution of a sample $X_1, \dots, X_n \mid \tilde{p} \overset{\text{iid}}{\sim} \tilde{p}$. From (1.3), it is clear that the realizations $\omega \mapsto \tilde{p}(\omega)$ from a DP are almost surely discrete. Therefore, with a positive probability, there will be ties among the X_i's. Let $X_1^*, \dots, X_k^*$ be the unique values in $(X_1, \dots, X_n)$ and $n_h = \#\{i : X_i = X_h^*\}$, then

$$\begin{aligned} \mathsf{P}(X_1 \in \mathrm{d}x_1, \dots, X_n \in \mathrm{d}x_n) = \mathsf{P}(n_1, \dots, n_k, X_1^* \in \mathrm{d}x_1^*, \dots, X_k^* \in \mathrm{d}x_k^*) = \\ \frac{\alpha^k}{(\alpha)_n} \prod_{h=1}^{k} (n_h - 1)! \prod_{h=1}^{k} G_0(\mathrm{d}x_h^*) \end{aligned} \tag{1.4}$$

where $(\alpha)_n = \alpha(\alpha+1) \cdots (\alpha+n-1)$, see, for instance, Antoniak (1974).

Note that the marginal distribution of the sample $(X_1, \dots, X_n)$ can be factored in two parts: one that depends exclusively on the unique values displayed (that is, $\prod_{h=1}^{k} G_0(\mathrm{d}x_h^*)$) and one that depends on the *partition* of the indices $\{1, \dots, n\}$ in *clusters* $\mathcal{C}_1, \dots, \mathcal{C}_k$ defined by $\mathcal{C}_j = \{i : X_i = X_j^*\}$. In (1.4) this second term corresponds to $\alpha^k / (\alpha)_n \prod_{h=1}^{k} (n_h - 1)!$ and depends specifically on the partition only through the sample size n and the cardinalities $n_1, \dots, n_k$. More generally, the marginal law of a sample from an almost surely discrete random probability measure with independent and identically distributed (i.i.d.) atoms from a diffuse measure G_0 can be factored into $\prod_{i=1}^{k} G_0(\mathrm{d}_k^*)$ times the prior distribution induced on the partition, usually termed *exchangeable partition probability function* (EPPF, Pitman, 1995) which can be understood as

$$\text{EPPF}(n_1, \dots, n_k) = \int_{\Theta^k} \mathbb{E}\left[\tilde{p}^{n_1}(\mathrm{d}x_1^*) \cdots \tilde{p}^{n_k}(\mathrm{d}x_k^*)\right].$$

As shown in Blackwell and MacQueen (1973), (1.4) can be interpreted in terms of a generalized Pólya urn, also referred to as the Chinese restaurant process (CRP Pitman, 2006) metaphor or Ewens' sampling formula (Ewens, 1972). In this metaphor, we imagine a Chinese restaurant with infinite tables, where customers enter one by one. At each table, only one dish τ_j is associated. Then, the first customer sits at the first table eating dish $\tau_1 \sim G_0$. The second customer sits at the first table with probability proportional to 1 and at the second ("new") table with probability proportional to α. If the second table is chosen, dish $\tau_2 \sim G_0$ is assigned to the table, independently of τ_1. After n customers have entered the restaurants, suppose that k tables have been selected with n_h, $h = 1, \dots, k$ customers sitting at each table. Of course, $\sum_h n_h = n$. Then, the $n+1$-th customer sits at table h with probability proportional to n_h, $h = 1, \dots, k$ or at a "new" table $k+1$ with probability proportional to α.

1.1.2 Gibbs type priors

The interpretation of the marginal law of a sample from a random probability measures as a generalized Pólya urn is not specific to the Dirichlet process. Indeed, Pitman (1996) proved that these predictive distributions characterize the large class of species sampling models. We review this fundamental result hereafter.

Let $(w_h)_{h \geq 1}$ be a sequence of nonnegative random variables such that $\sum_{h \geq 1} w_h \leq 1$ almost surely. Let $(\tau_h)_{h \geq 1}$ be a sequence of i.i.d. random variables from a non-atomic probability measure P_0. Then the random probability measure

$$\tilde{p}(\cdot) = \sum_{h \geq 1} w_h \delta_{\tau_h} + \left(1 - \sum_{h \geq 1} w_h\right) P_0 \tag{1.5}$$

is called a species sampling model. A sequence of random variables $(X_i)_{i \geq 1}$ such that $X_i \mid \tilde{p} \overset{\text{iid}}{\sim} \tilde{p}$ is called a species sampling sequence. As proven by Pitman (1996), species sampling sequences can be equivalently characterized in terms of the predictive distribution of X_{n+1} given $X_1, \dots, X_n$. Indeed $(X_i)_{i \geq 1}$ is a species sampling sequence if and only if there exists a sequence of i.i.d. random variables $(\tau_h)_{h \geq 1}$ from a nonatomic probability measure P_0 and a collection of weights $\{p_{h,n}(n_1, \dots, n_k) : 1 \leq h \leq k, \ 1 \leq k \leq n, \ n \geq 1\}$ such that $X_1 = \tau_1$ and

$$X_{n+1} \mid X_1, \dots, X_n = \begin{cases} \tau_{n+1}, & \text{with prob. } p_{k_n+1,n}(n_1, \dots, n_{k_n}, 1) \\ X^*_{h,n}, & \text{with prob. } p_{k_n,n}(n_1, \dots, n_h + 1, \dots, n_{k_n}) \end{cases} \tag{1.6}$$

where $X^*_{1,n}, \dots X^*_{k_n,n}$ denote the distinct values in $X_1, \dots, X_n$, each appearing with frequency n_h, $h = 1, \dots, k_n$, and k_n is their cardinality.

It is trivial to interpret the CRP in the form of (1.6) where $p_{k_n+1,n}(n_1, \dots, n_{k_n}, 1) = \alpha/(\alpha + n)$ and $p_{k_n,n}(n_1, \dots, n_h + 1, \dots, n_{k_n}) = n_h/(\alpha + n)$. In particular, Zabell (2005) proved that the Dirichlet process is the only species sampling model for which the probability of observing a "new" value for X_{n+1}, i.e., different from $X_1, \dots, X_n$, depends only on the previously observed sample through its cardinality. See also Bacallado et al. (2017). This lack of flexibility has been previously criticized. On the other hand, the functions $\{p_{h,n}\}$ are in general impossible to compute analytically when starting from the definition (1.5), which makes general species sampling models cumbersome for Bayesian inference.

Gibbs type priors (Gnedin and Pitman, 2005; Lijoi et al., 2008; De Blasi et al., 2013) provide a convenient trade-off between analytical tractability and flexibility. We say that a species sampling sequence $(X_i)_{i \geq 1}$ of Gibbs type if there exists a parameter $\sigma < 1$ and a triangular array of positive weights $\{V_{n,h}, \ 1 \leq h \leq n, \ n \geq 1\}$ satisfying the recursive relation

$$V_{n,h} = (n - \sigma h)V_{n+1,h} + V_{n+1,h+1}, \quad h = 1, \dots, n \; n = 1, 2, \dots$$

with $V_{1,1} = 1$, such that the probabilities in (1.6) are equal to

$$p_{k_n+1,n}(n_1, \dots, n_{k_n}, 1) = \frac{V_{n+1,k_n+1}}{V_{n,k_n}}$$

$$p_{k_n,n}(n_1, \dots, n_h + 1, \dots, n_{k_n}) = \frac{V_{n+1,k_n+1}}{V_{n,k_n}}(n_h - \sigma)$$

Apart from the DP, one of the most notable Gibbs type priors is the two-parameter Poisson-Dirichlet distribution or Pitman-Yor process (PYP Pitman and Yor, 1997), where

$$p_{k_n+1,n}(n_1, \dots, n_{k_n}, 1) = \frac{\theta + k_n\sigma}{\theta + n}, \quad p_{k_n,n}(n_1, \dots, n_h + 1, \dots, n_{k_n}) = \frac{n_j - \sigma}{\theta + n}$$

the parameter $\sigma \in (0, 1)$ can be interpreted as a "discount" parameter, controlling the reinforcement. Also the PYP admits a stick-breaking representation as in (1.3) (Ishwaran and James, 2001), where the random variables ν_j are independent and distributed as Beta$(1 - \sigma, \alpha + \sigma j)$.

1.1.3 Completely Random Measures and their normalization

A very fruitful approach to defining random probability measures has been through the normalization of random measures. This idea, introduced in Regazzini et al. (2003) for random probabilities on the real line with the name of normalized random measures with independent increments (NRMIs), has been extensively studied in the last two decades. See Lijoi and Prünster (2010) for an overview.

Before stating this construction, technical preliminaries are needed. The first building block is the Poisson process. See Kingman (1993); Daley and Vere-Jones (2003, 2008) for a detailed account. Let (Θ, d) denote a measurable space endowed with the Borel sigma algebra, and let λ be a non-null measure on Θ. We denote by N the Poisson random measure on Θ and write $N \sim \mathcal{P}(\lambda)$ to denote the law of a Poisson random measure with intensity λ. That is, $N \sim \mathcal{P}(\lambda)$ is a random counting measure such that for any collection of pairwise disjoint measurable sets $A_1, \dots, A_k$

$$\mathsf{P}(N(A_1) = n_1, \dots, N(A_k) = n_k) = \prod_{j=1}^{k} \frac{(\Lambda(A_j))^{n_j}}{n_j!} e^{-\Lambda(A_j)}$$

where $\Lambda(A) := \int_A \lambda(\theta)\mathrm{d}\theta$.

The Poisson process is an example of completely random measure (CRM, Kingman, 1967), that is a random measure ν such that the random variables $N(A_1), \dots, N(A_k)$ are independent for pairwise disjoint measurable sets A_j. Moreover, the Poisson process is also the fundamental building block for other kinds of completely random measures, as shown in Kingman (1967). Indeed, if a random measure μ on Θ has no fixed atoms (i.e., $\mathsf{P}(\nu(\{x\}) > 0) = 0$ for any x), it holds that

$$\mu(A) = \int_{\mathbb{R}_+ \times A} sN(\mathrm{d}s\mathrm{d}\theta)$$

where N is a Poisson random measures on the extended space $\mathbb{R}_+ \times \Theta$. Hence, to build a completely random measure on Θ, it is sufficient to consider the Poisson random measure on $\mathbb{R} \times \Theta$. With a slight abuse of notation, we denote with $\lambda(\mathrm{d}s\mathrm{d}\theta)$ the intensity measure of μ. When $\lambda(\mathrm{d}s\mathrm{d}\theta) = \rho(\mathrm{d}s)\alpha(\mathrm{d}x)$, where ρ is a measure on $\mathbb{R}_+$ and α is a finite measure on Θ, the random measure is said to be *homogeneous*.

Given a random measure μ on Θ, it is natural to consider a random probability measure by setting, for measurable sets A

$$\tilde{p}(A) = \frac{\mu(A)}{\mu(\Theta)}.$$

However, some care must be taken to ensure that $\tilde{p}$ is well defined. Sufficient conditions that ensure $\mathsf{P}(\mu(\Theta) = 0) = 0$ and $\mathsf{P}(\mu(\Theta) < +\infty) = 1$ are

$$\int_{\mathbb{R}_+} \rho(\mathrm{d}s) = +\infty, \qquad \int_{\mathbb{R}_+} \min\{1, s\}\rho(\mathrm{d}s) < +\infty;$$

see Regazzini et al. (2003) for further details. Observe that the number of atoms in $\tilde{p}$ is unbounded thanks to the condition $\int \rho(\mathrm{d}s) = +\infty$. From the Poisson process representation, we can further see that the atoms of $\tilde{p}$ are i.i.d. from a probability distribution $G_0(\mathrm{d}\theta) = \alpha(\mathrm{d}\theta)/\alpha(\Theta)$.

Notable examples of NRMIs are the Dirichlet process, which obtained by normalizing a Gamma random measure, for which $\lambda(\mathrm{d}s\mathrm{d}\theta) = s^{-1}e^{-s}\mathrm{d}s\gamma G_0(\mathrm{d}\theta)$ where $\gamma > 0$ and G_0 is a probability measure over Θ; the normalized stable process, where $\lambda(\mathrm{d}s\mathrm{d}\theta) = \sigma s^{-1-\sigma}/\Gamma(1-\sigma)\mathrm{d}s\gamma G_0(\mathrm{d}\theta)$ for $\sigma \in (0, 1)$; the normalized inverse Gaussian process (Lijoi et al., 2005); the normalized generalized Gamma process (Brix, 1999; Lijoi et al., 2007). Specifically, the normalize generalized Gamma process corresponds to setting

$$\lambda(\mathrm{d}s\mathrm{d}\theta) = \frac{e^{-\theta s}}{\Gamma(1-\sigma)s^{1+\sigma}}\mathrm{d}s\, cG_0(\mathrm{d}\theta)$$

and includes as special cases the Dirichlet process ($\theta = 1$, $\sigma = 0$), the normalized inverse Gaussian process ($\sigma = \theta = 1/2$, $c = b^{1/2}/\sqrt{2}$), and the normalized stable process ($\theta = 0$, $c = \gamma\sigma$). As shown in Pitman and Yor (1997), the Pitman-Yor random probability measure cannot be obtained via normalization of a CRM, but rather via the normalization of a suitable transformation of the stable CRM.

1.1.4 Inference with Nonparametric Mixtures

The almost sure discreteness of $\tilde{p}$, makes it cumbersome to assume the DP, PYP or any NRMI as the prior distribution Q in (1.1) when observations $X_1, \ldots, X_n$ are assumed to be continuous. Nonparametric mixtures have been first introduced in Ferguson (1983) and Lo (1984). It was Escobar and West (1995), where a simulation algorithm to approximate the posterior algorithm was developed, that made it become common practice to use the DP (and other almost surely discrete random probability measures) as the prior distribution for the mixing measure in a mixture model. That is, the model for observations $Y_1, \ldots, Y_n$ is

$$\begin{aligned} Y_1, \ldots, Y_n \mid \tilde{p} &\overset{\text{iid}}{\sim} \int_\Theta f(\cdot \mid \theta)\tilde{p}(\mathrm{d}\theta) \\ \tilde{p} &\sim Q \end{aligned} \tag{1.7}$$

where $\{f(\cdot \mid \theta)\}_{\theta\in\Theta}$ is a parametric family of densities (with respect to Lebesgue measure on $\mathbb{R}^q$ or counting measure on a countable subset of $\mathbb{R}^q$), and the specification of the parameter space Θ is depends on the application. We refer to this parametric family as the kernel (of the mixture model). The most popular example consists in letting $\theta = (\mu, \sigma^2)$ and $f(\cdot \mid \theta) = \mathcal{N}(\cdot \mid \mu, \sigma^2)$, where, with an abuse of notation, we use $\mathcal{N}(\cdot \mid \mu, \sigma^2)$ to represent the probability density function of a Gaussian random variable with mean μ and variance σ^2.

Given the extensive literature on Bayesian nonparametric mixture models, we limit ourselves to cite two fundamental papers: Neal (2000), where several efficient algorithms for Dirichlet process mixtures have been introduced, and James et al. (2009), where the authors extended the class of BNP mixutres to NRMIs mixing distributions, through the study of the EPPF induced by NRMIs and their posterior representation.

From (1.7), it is clear that one can estimate the data generating density by the posterior predictive distribution:

$$p(y \mid y_1, \ldots, y_n) = \int \left(\int_\Theta f(y \mid \theta) \tilde{p}(\mathrm{d}\theta) \right) \Pi(\mathrm{d}\tilde{p} \mid y_1, \ldots, y_n),$$

where $\Pi(\mathrm{d}\tilde{p} \mid y_1, \ldots, y_n)$ is the posterior distribution of $\tilde{p}$ given the data. Since this posterior is not available in closed form, we can employ a Monte Carlo approximation to the integral. That is, given samples $\tilde{p}^{(1)}, \ldots, \tilde{p}^{(M)}$ from $\Pi(\mathrm{d}\tilde{p} \mid y_1, \ldots, y_n)$, obtained, for instance, via a Markov chain Monte Carlo algorithm, we have

$$p(y \mid y_1, \ldots, y_n) \approx \widehat{p}(y) = \frac{1}{M} \sum_{m=1}^{M} \int_\Theta f(y \mid \theta) \tilde{p}^{(m)}(\mathrm{d}\theta),$$

where M is large.

Mixture models are also a popular framework for model-based clustering. Indeed, we can reformulate (1.7) as the following hierarchical model

$$\begin{aligned} Y_i \mid \theta_i &\overset{\text{ind}}{\sim} f(\cdot \mid \theta_i), \qquad i = 1, \ldots, n \\ \theta_i \mid \tilde{p} &\overset{\text{iid}}{\sim} \tilde{p} \qquad i = 1, \ldots, n \\ \tilde{p} &\sim Q \end{aligned}$$

The almost-sure discreteness of $\tilde{p}$ entails that $\mathsf{P}(\theta_i = \theta_j) > 0$. Therefore, we can partition observations into clusters based on the latent θ_j's.

1.2 Why atoms' interaction?

In this section, we will argue in favor (*i*) finite mixture models and (*ii*) introducing a dependence across the support points in $\tilde{p}$.

A finite mixture model has the form (1.7) where $\tilde{p} = \sum_{h=1}^{m} w_h \delta_{\theta^*_h}$, with $m < +\infty$ almost surely. The random elements defining $\tilde{p}$ are then $\{\theta^*_h\}_h$, $\boldsymbol{w} = (w_1, \ldots, w_m)$ and possibly m.

Historically, infinite mixture models have been preferred to the finite ones thanks to the availability of efficient Markov chain Monte Carlo (MCMC) algorithms to perform posterior inference. Indeed, starting from the seminal work of Neal (2000) several algorithms have been proposed to fit nonparametric mixture models. Most importantly, these algorithms are suitable for all choices of mixture kernel $f(\cdot \mid \theta)$ and base measure G_0. Instead, traditional approaches to deal with a random number of components m have been based on reversible jump MCMC (Green, 1995). In this case, the algorithm must be tailored to the choices of f and G_0 and efficiency of the algorithm is usually a concern.

The inconsistency of the Pitman-Yor mixture model for the number of clusters, proven in Miller and Harrison (2014a), led to reconsidering finite mixture models with m random as a suitable alternative to infinite mixtures for model-based clustering. Informally, Miller and Harrison (2014a) proved that, if the true data generating process consists of a mixture of k_0 components $f(\cdot \mid \theta^*_1), \ldots, f(\cdot \mid \theta^*_{k_0})$, then the posterior distribution of the number of clusters k in a Pitman-Yor mixture model does not converge to δ_{k_0} when the number of observations grows to infinity. At the same time, the recent works Miller and Harrison (2018), Argiento and De Iorio (2022), and Frühwirth-Schnatter et al. (2021) established several connections between finite mixture models with m random and mixture models

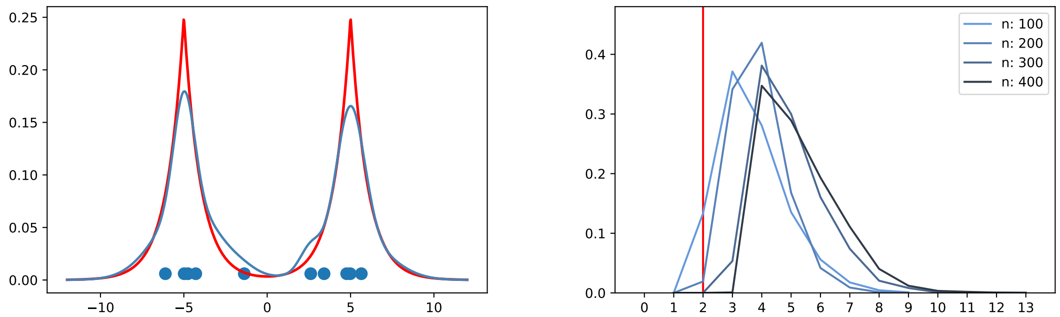


Figure 1.1: Left: true data generating density (red), Bayesian density estimate (blue), estimated cluster centres (blue dots). Right: posterior distribution of the number of clusters, for different sample sizes n.

where $m = +\infty$, developing efficient and general algorithms inspired by Neal (2000) for finite mixture models.

It has been known since Nobile (1994) that finite mixture models consistently estimate m (in turn, this guarantees the consistency of the estimated number of clusters), provided that the model is well-specified. The main assumption, which cannot be verified in practice, is that the mixture kernel $f(\cdot \mid \cdot)$ agrees with the true data generating process. If this is not the case, then finite mixture models cannot be expected to be consistent for m. In particular, one can easily imagine that if the mixture kernel does not agree with the true data generating process, the mixture model will estimate a large number of components m to faithfully approximate the data generating density. This point has been theoretically addressed in Cai et al. (2021) in the limit of the sample size growing to infinity.

Let us give a practical example of this behavior, showing that the inconsistency occurs even with modest sample sizes. We generated n observations from a mixture of two Laplace distributions, located respectively in -5 and $+5$. Figure 1.1 shows the true data generating density (left plot, red) and the density estimate obtained with a location-scale mixture of Gaussians. Note that the density estimate is extremely precise, especially near the centers of the two components. However, in order to obtain such a good estimate, the number of clusters estimated is almost always greater than the “true” number of clusters, that is two (right plot).

This simple example shows an imbalance. Mixture models are used routinely for density estimation and clustering, but they clearly tend to favor density estimation accuracy over clustering. In practice, especially when the data dimension is greater than two, it is almost impossible to recognize that a mixture model is misspecified since posterior summaries such as the density estimate cannot be easily visualized.

In the next chapters, we set out to pursue the opposite trade-off: favoring interpretable clustering over density estimation. The approach is based on forcing separation among the components’ densities. Specifically, by assuming a (repulsive) point process distribution as prior for the cluster centers and the number of components m.

1.3 Previous work on repulsive mixture models

Repulsive mixtures have been previously considered in Petralia et al. (2012), Xu et al. (2016), Fúquene et al. (2019), Quinlan et al. (2020), Bianchini et al. (2020), and Xie and Xu (2019). In these papers, the mixture kernel $f(\cdot \mid \cdot)$ in (1.7) is assumed to be the Gaussian density with parameters (μ, γ), which represent the cluster-specific mean and variance (if data are univariate) or covariance matrix (if data are multivariate). Then, a joint prior $p(\boldsymbol{\mu})$, where $\boldsymbol{\mu} = \{\mu_1, \dots, \mu_m\}$, is assumed for the cluster-specific means, including possibly the cardinality m.

In particular, in Petralia et al. (2012), Fúquene et al. (2019), and Quinlan et al. (2020), m is finite and fixed, but this cannot guarantee the posterior consistency of the number of components. However, Xu et al. (2016), Bianchini et al. (2020), and Xie and Xu (2019) assume m to be finite and random. In particular, Xu et al. (2016) and Bianchini et al. (2020) deal with determinantal point process (DPP) priors for $\boldsymbol{\mu}$. A DPP density has to be approximated as described in Lavancier et al. (2015) where the computational complexity will increase exponentially fast as the dimension q increases, or as described in Bardenet and Titsias (2015) (but at the price that the model parameters are hard to interpret). Xie and Xu (2019) assume a tempered repulsive pairwise interaction point process density as prior for $\boldsymbol{\mu}$ conditioned on m:

$$p(\mu_1, \dots, \mu_m \mid m) = \frac{1}{Z_m} \left[\prod_{i=1}^{m} \phi_1(\mu_i) \right] \left[\prod_{1 \leq i < j \leq m} \phi_2(\|\mu_i - \mu_j\|)^{1/m} \right]$$

with respect to m-fold Lebesgue measure on $\mathbb{R}^q$. Here, $\| \cdot \|$ denotes usual distance, ϕ_1 is a non-negative function, $0 \leq \phi_2 \leq 1$ is a non-decreasing function (this implies repulsiveness), and Z_m is the normalizing constant. Note that if $\phi_2(\cdot) = 1$, then $\mu_1, \dots, \mu_m$ are iid and independent of m. Apart from this case, Z_m is intractable and has to be approximated by numerical methods, a non-trivial task which limits both efficiency and feasibility as the dimension q increases.

As far as posterior simulation is concerned, Xu et al. (2016) and Bianchini et al. (2020) proposed to simulate $(\boldsymbol{w}, \boldsymbol{\tau})$ using a reversible jump MCMC algorithm, cf. Green (1995). At every iteration of this algorithm, either a split move (in which one component is killed and two new ones are created, hence increasing the dimension by one), or a combine move (in which two components are merged into a single one, hence decreasing the dimension by one) is proposed. As discussed in Green (2010), Richardson and Green (1997), and Dellaportas and Papageorgiou (2006), in order to obtain good mixing properties of the reversible jump MCMC algorithm, it is crucial to define appropriate proposal distributions that generate the new values in the split move. In general, this is a complex task that depends heavily on the kernel under consideration.

Similarly to how Miller and Harrison (2018) studied a classical mixture model, Xie and Xu (2019) consider in the observation model (1.7) to marginalize with respect to a prior of $(\boldsymbol{w}, \boldsymbol{\tau})$ and derive a ‘marginal MCMC algorithm’. However, although this algorithm compared with the reversible jump MCMC algorithm has smaller auto-correlations for the number of clusters, it requires the calculation of the normalizing constants $Z_1, Z_2, \dots$ up to some truncation, and inference is limited to the number of clusters and the posterior mean of the mixture density.

2. MCMC computations for Bayesian mixture models using repulsive point processes

In this chapter, based on Beraha et al. (2022), we consider mixture models of the kind (1.7), where m is finite and random. The focus here is to extend the approach in Argiento and De Iorio (2022), considering in particular prior specification and Bayesian MCMC computations when the aim is cluster detection. Our objective is partly to present a general framework for mixture models based on repulsive point process priors for 'cluster centers', arguing why this is useful, and partly to derive a MCMC algorithm which avoids the well-known difficulties associated with reversible jump MCMC computation. In several simulation studies and an application on sociological data, we illustrate the advantage of our new methodology over existing methods, and we compare the use of the different repulsive point process priors. Moreover, when introducing a hyperparameter in such priors, we demonstrate that perfect simulation is fast in connection to a useful ancillary variable method.

2.1 Setting

For specificity, assume each $y_i \in \mathbb{R}^q$ with $q \geq 1$. It will always be obvious from the context whether we consider y_i (and other variables considered later on) as a random variable, a realization, or an argument of a function. In this chapter, we specialize (1.7) as

$$y_i \mid \boldsymbol{w}, \boldsymbol{\tau} \overset{\text{iid}}{\sim} \sum_{h=1}^{m} w_h f(\cdot \mid \tau_h), \qquad i = 1, \dots, n. \tag{2.1}$$

The densities $f(\cdot \mid \tau_h)$, $h = 1, \dots, m$ are usually referred to as the 'components' of the mixture. In this context, cluster detection means estimating the allocation parameters $\boldsymbol{c} = (c_1, \dots, c_n) \in \{1, \dots, m\}^n$ where the sets $\{y_i : c_i = h\}$, $h = 1, \dots, m$ are the clusters. The number of clusters in the mixture model is the number of allocated components in (2.1), i.e., the number of unique values in $(c_1, \dots, c_n)$.

We make prior assumptions as follows. To control the number of clusters, m is random and finite; the case $m = +\infty$ would be relevant for nonparametric inference (Müller and Mitra, 2013), but this context is not addressed in this chapter. Only when $m < +\infty$ is not fixed, it can be consistently estimated, cf. Argiento and De Iorio (2022) and Miller and Harrison (2018). We let $\tau_h = (\mu_h, \gamma_h)$, thinking of μ_h as a continuous random parameter in $\mathbb{R}^q$ that specifies a 'cluster center' of cluster h, and of γ_h as a positive random parameter ($q = 1$) or a continuous covariance matrix ($q \geq 2$) (or, in simple settings, a fixed positive number) which specifies the amount of dispersion of the data points in cluster h (for example, $f(\cdot \mid \tau_h)$ could be a normal density with mean μ_h and variance γ_h). To make posterior inference more robust, we add a hyperparameter ξ to the prior distribution of $(\mu_1, \dots, \mu_m)$. Furthermore, since the mixture density in (2.1) does not depend on the order of the components, we can assume that

(a) the conditional marginal prior density $p(\mu_1, \dots, \mu_m \mid \xi, m)$ is exchangeable, that is, for any fixed integer $m \geq 1$, it is invariant under permutations of $\mu_1, \dots, \mu_m$. Note that $\boldsymbol{\mu} = \{\mu_1, \dots, \mu_m\}$ is then a finite point process, specifying both the random number m of components and the locations of the cluster centers. Finally, a priori we make conditional independence assumptions: Conditioned on (ξ, m), we have that

(b) $(w_1, \dots, w_m)$, $(\mu_1, \dots, \mu_m)$, and $(\gamma_1, \dots, \gamma_m)$ are a priori independent,

(c) given m, the conditional marginal prior distribution of $(w_1, \dots, w_m)$ does not depend on ξ,

(d) the γ_h's are iid, with a prior distribution which does not depend on (ξ, m),

and conditioned on $(\xi, m, w_1, \dots, w_m, \mu_1, \dots, \mu_m, \gamma_1, \dots, \gamma_m)$, we have that

(e) the c_i's are iid with a prior distribution given by $P(c_i = h \mid \boldsymbol{w}) = w_h$.

Hence, the random parameter here consists of $(\xi, \{\mu_1, \dots, \mu_m\}, w_1, \dots, w_m, \gamma_1, \dots, \gamma_m,$ $c_1, \dots, c_n)$. By Bayes' theorem, using the generic notation $p(\cdot)$ for a density and $p(\cdot \mid \cdot)$ for a conditional density, the posterior density becomes

$$\begin{aligned} p(\xi, \{\mu_1, \dots, \mu_m\}, & w_1, \dots, w_m, \gamma_1, \dots, \gamma_m, c_1, \dots, c_n \mid y_1, \dots, y_n) \propto \\ & p(\xi) p(m \mid \xi) p(\mu_1, \dots, \mu_m \mid \xi, m) p(w_1, \dots, w_m \mid m) \\ & \left[\prod_{h=1}^{m} p(\gamma_h)\right] \left[\prod_{i=1}^{n} w_{c_i} f(y_i \mid (\mu_{c_i}, \gamma_{c_i}))\right]. \end{aligned} \tag{2.2}$$

The dominating measure for (2.2) is given in Section 2.5 which contains measure theoretical details; see Section 2.3 for further prior specifications. In brief, the prior specification of $\boldsymbol{\mu}$ and $\boldsymbol{w}$ requires particular attention, whilst for the prior specification of the remaining parameters we use a standard setting, following Fraley and Raftery (2007).

Note that we impose the hyperprior on ξ, the parameter in the repulsive point process prior controlling the intensity of the point process, to make posterior inference more robust, cf. Section 2.1, unlike previous literature (apart from Bianchini et al., 2020). When making posterior updates of ξ in our MCMC algorithm, if the prior density for $\boldsymbol{\mu}$ conditioned on ξ has an intractable normalizing constant Z_ξ, we get rid of Z_ξ by using the single exchange algorithm in Murray et al. (2006) coming from the ancillary variable algorithm in Møller et al. (2006). These algorithms require perfect simulation of an auxiliary variable following the same distribution as $\boldsymbol{\mu}$ conditioned on ξ. Interestingly, perfect simulation is feasible in our context because m will typically be small (in our examples, it is effectively always less than 10).

2.2 Our Contribution and Outline

We discuss a general framework for mixture models based on repulsive point process priors for 'cluster centers' $\boldsymbol{\mu}$ and derive a new MCMC algorithm avoiding the problem with reversible jump MCMC computation.

Our first contribution is the proposal of the prior of $\boldsymbol{\mu}$ conditioned on ξ, cf. item (a) in Section 2.1: We consider a general setting with a repulsive finite point process density, including the case of a DPP (any DPP except the special case of a Poisson process is repulsive) or a density specified by an unnormalized density, e.g. a pairwise interaction point process density, which involves a normalizing constant Z_ξ which in general (except the special case of a Poisson process) is intractable. As a particular simple example of a pairwise interaction point process, we assume a Strauss process (defined later in Section 2.3.1). Note that the prior distributions for $\boldsymbol{\mu}$ in all the papers cited in Section 1.3

can all be considered as special cases of our prior for $\boldsymbol{\mu}$. Also note that Z_ξ will never appear in our posterior simulation algorithm.

The second contribution is the algorithm for posterior simulation from our model. This contribution builds upon Argiento and De Iorio (2022) and is mainly based on two assumptions, namely $\boldsymbol{w}$ and $\boldsymbol{\mu}$ are chosen a priori independent and the mixture weights $\boldsymbol{w}$ are defined by *normalization* of iid infinitely divisible random variables, i.e. $\boldsymbol{w}$ follows a *normalized infinitely divisible* distribution (Favaro et al., 2011). In fact, Argiento and De Iorio (2022) introduced the class of *normalized independent point processes mixture* models and showed that this class can be framed in the nonparametric Bayesian context. In this way, several ideas and algorithms developed in the nonparametric literature for normalized random measures with independent increments (NRMI – see Regazzini et al., 2003) can be adapted to the finite-dimensional case. Here we extend Argiento and De Iorio (2022) building a Metropolis-within Gibbs sampler, referred to as *conditional Gibbs sampler* in the Bayesian nonparametric literature; see Papaspiliopoulos and Roberts (2008). In particular, we relax the usual assumption of $\mu_1, \dots, \mu_m$ being iid and independent of m, still being able to propose a transformation of $\boldsymbol{\mu}$ into allocated cluster centers $\boldsymbol{\mu}^{(a)} = \{\mu_{c_i} : i = 1, \dots, n\}$ and non-allocated cluster centres $\boldsymbol{\mu}^{(na)} = \boldsymbol{\mu} \backslash \boldsymbol{\mu}^{(na)}$. This allows us to simulate from the full conditional of $\boldsymbol{\mu}$ without resorting to the split and combine moves of the reversible jump MCMC algorithm as used in Xu et al. (2016) and Bianchini et al. (2020). In fact, posterior updates of $\boldsymbol{\mu}^{(a)}$ become easy and when updating $\boldsymbol{\mu}^{(na)}$ we use the Metropolis-Hasting birth-death algorithm in Geyer and Møller (1994). The Metropolis-Hasting birth-death algorithm has the advantage that the choice of the kernel does not impact the acceptance rate of the algorithm.

The remainder of this chapter is organized as follows. Sections 2.3 and 2.4 specify our further prior assumptions on the cluster centers $\boldsymbol{\mu}$ and the mixture weights $\boldsymbol{w}$, respectively. Section 2.5 derives the posterior density, using the useful superposition of $\boldsymbol{\mu}$ mentioned above, and provides the technical details needed when dealing with point process densities (we aim at keeping this as simple as possible). Section 2.6 details our Metropolis-within-Gibbs sampler for posterior simulation. Sections 2.7.1 and 2.7.2 discuss prior elicitation when the prior for $\boldsymbol{\mu}$ is the Strauss process and the DPP (conditioned on $\{m \geq 1\}$, see Section 2.3). Section 2.8 presents various simulation studies comparing posterior inference and MCMC mixing obtained using reversible jump or our Metropolis-within-Gibbs sampler, and using a DPP, a Strauss process, or a non-repulsive prior for $\boldsymbol{\mu}$. Furthermore, an application to a sociological data set is discussed in Section 2.9. The article concludes with a discussion in Section 2.10. In the Appendix we provide practical details on our Metropolis-within-Gibbs sampler, collect additional simulation studies, including an illustration on the advantages of using a Strauss process over a DPP as prior for $\boldsymbol{\mu}$, and discuss possible extensions.

2.3 Prior specification of the cluster centers

For the prior specification of $\boldsymbol{\mu}$, introduced in item (a) in Section 2.1, which is the first original contribution of our work, a few technical details are needed to start. Let $\Omega = \cup_{m=0}^{\infty} \Omega_m$ denote the space of all finite subsets (point configurations) of $\mathbb{R}^q$, where Ω_m denotes the space of all finite subsets of cardinality m, with $\Omega_0 = \{\emptyset\}$, where $\emptyset$ denotes the empty point configuration; although we cannot have 0 groups, it becomes convenient in Section 2.5 to include Ω_0 into the definition of Ω. We equip each Ω_m with the smallest σ-algebra making the mapping of pairwise distinct $(\mu_1, \dots, \mu_m) \in \mathbb{R}^{qm}$ into $\{\mu_1, \dots, \mu_m\} \in \Omega_m$ measurable. The σ-algebra on Ω is the smallest σ-algebra that contains the union of the σ-algebras on each Ω_m. Then $\boldsymbol{\mu}$ is absolutely continuous with respect to a measure on Ω which, with an abuse of notation, is denoted $\mathrm{d}\boldsymbol{\mu}$ and defined as follows. For sets

$B = \cup_{m=0}^{\infty} B_m$ with $B_m \subseteq \Omega_m$,

$$\int_B \mathrm{d}\boldsymbol{\mu} = \sum_{m=0}^{\infty} \frac{1}{m!} \int_{B_m} \mathrm{d}\boldsymbol{\mu}_m,$$

where the notation means the following. For $m = 0$, we interpret the term in the sum as $[\emptyset \in A]$. We set $\boldsymbol{\mu}_m = \{\mu_1, \ldots, \mu_m\}$ and, with an abuse of notation, write $\mathrm{d}\boldsymbol{\mu}_m$ for Lebesgue measure $\mathrm{d}\mu_1 \cdots \mathrm{d}\mu_m$ on $\mathbb{R}^{qm}$. Further, we write $\int_{B_m} \mathrm{d}\boldsymbol{\mu}_m$ for $\int_{\mathbb{R}^{qm}} \mathbb{I}[\boldsymbol{\mu} \in B_m]\, \mathrm{d}\boldsymbol{\mu}_m$, where $\mathbb{I}[\cdot]$ denotes the indicator function. Then, conditioned on ξ, the density of $\boldsymbol{\mu}$ with respect to $\mathrm{d}\boldsymbol{\mu}$ is given by

$$p(\boldsymbol{\mu} \,|\, \xi) = p(m \,|\, \xi) p(\mu_1, \ldots, \mu_m \,|\, \xi, m), \qquad \boldsymbol{\mu} = \{\mu_1, \ldots, \mu_m\} \in \Omega,\ m \geq 1,$$

setting $p(\mu_1, \ldots, \mu_m \,|\, \xi, m) = 0$ if $m = 0$. This means that we consider the prior process prior restricted to the event that $\boldsymbol{\mu}$ is non-empty.

2.3.1 Repulsive pairwise-interaction point process priors

When incorporating repulsiveness in the prior density $p(\boldsymbol{\mu} \,|\, \xi)$, we suggest a repulsive pairwise-interaction point process. This is a popular class of models in statistical physics and spatial statistics; see Møller and Waagepetersen (2004) and the references therein. The repulsive pairwise-interaction density is of the form

$$p(\boldsymbol{\mu} \,|\, \xi) = \frac{1}{Z_\xi} \left[\prod_{h=1}^{m} \phi_1(\mu_h \,|\, \xi) \right] \left[\prod_{1 \leq i < j \leq m} \phi_2(\|\mu_i - \mu_j\| \,|\, \xi) \right], \tag{2.3}$$

where $\phi_1(\cdot \,|\, \xi) \geq 0$ is an integrable function, $0 \leq \phi_2(\cdot \,|\, \xi) \leq 1$ is a non-decreasing function, and Z_ξ is a normalizing constant. Note that $Z_\xi < +\infty$, but in general Z_ξ is intractable. An exception is the special case $\phi_2(\cdot \,|\, \xi) = 1$ (a Poisson process with intensity function $\phi_1(\cdot \,|\, \xi)$ and conditioned on not being empty), where $Z_\xi = 1 - \exp(-\int \phi_1(\mu_h \,|\, \xi)\, \mathrm{d}\mu_h)$.

For simplicity and specificity, in Sections 2.8-2.9, we follow Bianchini et al. (2020) in letting ξ be a positive random variable and using an empirical Bayesian approach with

$$\phi_1(\mu_h \,|\, \xi) = \xi\, \mathbb{I}[\mu_h \in R], \tag{2.4}$$

where $R \subset \mathbb{R}^q$ is the smallest rectangular region containing the data $\boldsymbol{y}$ and with sides parallel to the usual axes in $\mathbb{R}^q$ (they advocate the use of this choice over other more complicated situations).

The simplest non-trivial case is a Strauss prior,

$$\phi_2(r \,|\, \xi) = \alpha^{\mathbb{I}[r \leq \delta]}, \tag{2.5}$$

so that ξ enters only in the expression of ϕ_1. Here, $\delta > 0$ is a fixed parameter, specifying the range of interaction, and $0 \leq \alpha \leq 1$ is a fixed interaction parameter. Note that, if $\alpha = 0$, we set $0^0 = 1$ and obtain a so-called hard core point process. If $\alpha = 1$, we obtain a model with no interaction which is like a Poisson process except that we condition on that $\boldsymbol{\mu}$ is non-empty.

2.3.2 Repulsive priors specified by an unnormalized density

In the following we consider a general prior model given by

$$p(\boldsymbol{\mu} \,|\, \xi) = \frac{1}{Z_\xi} g(\boldsymbol{\mu} \,|\, \xi), \tag{2.6}$$

where $g(\cdot\,|\,\xi)$ is a so-called unnormalized density, meaning that $g(\cdot\,|\,\xi)$ is a non-negative measurable function such that the normalizing constant Z_ξ is finite. Note that by assumption $g(\emptyset\mu\,|\,\xi) = 0$. Specific examples of (2.6) can be found in Møller and Waagepetersen (2004) and the references therein. In our simulation study and application example (Sections 2.8-2.9) we focus on the Strauss prior and a specific DPP prior given below, but considering (2.6) is useful in order to give a general exposition of our methodology.

To describe interaction in the general model (2.6), one possibility is to assume that for any $\boldsymbol{\mu} \in \Omega$ and $\mu^* \in \mathbb{R}^q \setminus \boldsymbol{\mu}$ we have $g(\boldsymbol{\mu} \cup \{\mu^*\}\,|\,\xi) > 0 \Rightarrow g(\boldsymbol{\mu}\,|\,\xi) > 0$, and then consider the so-called Papangelou conditional intensity defined by

$$\lambda(\mu^*, \boldsymbol{\mu}\,|\,\xi) := g(\boldsymbol{\mu} \cup \{\mu^*\}\,|\,\xi)/g(\boldsymbol{\mu}\,|\,\xi)$$

(taking $0/0 := 0$). Then we have repulsiveness if $\lambda(\mu^*, \boldsymbol{\mu}\,|\,\psi)$ is a non-increasing function of $\boldsymbol{\mu}$, that is, $\lambda(\mu^*, \boldsymbol{\mu}\,|\,\psi) \geq \lambda(\mu^*, \boldsymbol{\mu} \cup \{\mu'\}\,|\,\psi)$ for any $\mu' \in \mathbb{R}^q \setminus \boldsymbol{\mu} \cup \{\mu^*\}$, where inequality can not be replaced by an identity. Clearly, this is true for (2.3) when $\phi_2(\cdot\,|\,\xi) \neq 1$.

2.3.3 Determinantal point process priors

A DPP density (conditioned on that the DPP is non-empty) is a special case of (2.6) but with repulsion characterized in another way than above (Hough et al., 2009; Lavancier et al., 2015; Biscio et al., 2016; Møller and O'Reilly, 2021). To work with a DPP density, we consider a compact region $R \subset \mathbb{R}^q$ with $\int_R \mathrm{d}x > 0$, and a complex covariance function $C : R \times R \mapsto \mathbb{C}$ with a spectral representation

$$C(x, x'\,|\,\xi) = \sum_{i=1}^{\infty} \lambda_i \varphi_i(x)\overline{\varphi_i(x')}, \qquad x, x' \in R, \tag{2.7}$$

where the φ_i's form an orthonormal basis for the $L^2(R)$-space of complex functions defined on R, each $\lambda_i \geq 0$, and $\sum_{i=1}^{\infty} \lambda_i < +\infty$. Then the existence of the DPP is equivalent to that all $\lambda_i \leq 1$, cf. Macchi (1975). Note that we suppress in the notation that the eigenvalues λ_i's and the eigenfunctions φ_i's may depend on ξ.

A special case of a DPP occurs when C is a projection of finite rank m, let us say

$$C(x, x'\,|\,\xi) = \sum_{i=1}^{m} \varphi_i(x)\overline{\varphi_i(x')}, \qquad x, x' \in R.$$

From (2.7) we obtain a density

$$p(\mu_1, \ldots, \mu_m\,|\,\xi) = \det\{C(\mu_h, \mu'_h)\}_{h,h'=1,\ldots,m} \qquad \text{for } \mu_1, \ldots, \mu_m \in R, \tag{2.8}$$

where $\det\{C(\mu_h, \mu'_h)\}_{h,h'=1,\ldots,m}$ is the determinant of the $m \times m$ matrix $\{C(\mu_h, \mu'_h)\}_{h,h'=1,\ldots,m}$. A point process with density (2.8) is called a projection DPP with kernel C. Note that it consists of exactly m points in R.

The general construction of a DPP is given by introducing a random projection

$$K(x, x'\,|\,\xi) = \sum_{i=1}^{\infty} B_i \varphi_i(x)\overline{\varphi_i(x')}, \tag{2.9}$$

where $B_1, B_2, \ldots$ are independent Bernoulli variables with means $\lambda_1, \lambda_2, \ldots$, respectively. Then a DPP with kernel C is a finite point process on R which conditioned on $B_1, B_2, \ldots$ is a projection DPP with kernel K; it can be shown that the distribution of this DPP depends only on C, cf. Hough et al. (2006). Note that $\sum_{i=1}^{\infty} B_i$ is the random number of points. In particular, assuming all $\lambda_i < 1$ and defining C' as C in (2.7) but with each λ_i replaced by $\lambda_i' = \lambda_i/(1-\lambda_i)$, the DPP has unnormalized density

$$g(\boldsymbol{\mu} \,|\, \xi) = \det\{C'(\mu_h, \mu_{h'})\}_{h,h'=1,\dots,m} \qquad \text{for pairwise distinct } \boldsymbol{\mu} = \{\mu_1, \dots, \mu_m\} \subset R, m \geq 1. \tag{2.10}$$

Most DPP densities are specified as in (2.10) with the kernel coming from a parametric family of (often real) covariance functions with all eigenvalues < 1, see Lavancier et al. (2015). The advantage of using such models is that we can avoid including the Bernoulli variables as ancillary variables in the posterior, whilst the problem is to find a spectral representation. Note that when we condition on that the DPP is non-empty, the normalizing constant is given by

$$Z_\xi = \prod_{i=1}^{\infty} (1-\lambda_i)^{-1} - 1. \tag{2.11}$$

For our purpose it is easiest to let R be rectangular and use a spectral approach with Fourier basis functions for the eigenfunctions, cf. Lavancier et al. (2015). In Sections 2.8-2.9, we follow Bianchini et al. (2020) in making this choice of eigenfunctions and letting $R = [-\frac{1}{2}, \frac{1}{2}]^q$ and

$$C(x, x' \,|\, \xi) = \sum_{j \in \mathbb{Z}^q} \lambda_j \cos(2\pi j \cdot (x - x')), \tag{2.12}$$

where $\mathbb{Z}$ is the set of integers, $\cdot$ denotes the usual inner product on $\mathbb{R}^q$, and $\lambda_j = \chi(j)$ is specified by the spectral density χ of the power exponential spectral model from Lavancier et al. (2015). Specifically,

$$\lambda_j = \xi \frac{\alpha^q \Gamma(q/2+1)}{\pi^{q/2}\Gamma(q/\beta+1)} \exp(-\|\alpha j\|^\nu), \tag{2.13}$$

where $\Gamma(\cdot)$ is the gamma function, α and β are fixed positive parameters, and λ_j depends on the parameter $\xi > 0$ so that $\lambda_j \leq 1$ and $\sum_{j\in\mathbb{Z}^q} \lambda_j < \infty$. For details, see Lavancier et al. (2015), noting that ξ is the intensity of the DPP if we do not condition on that the DPP is non-empty.

When dealing with computations, in the sum of (2.12) and in the corresponding product $\prod_{j\in\mathbb{Z}^q} \cdots$ for the normalizing constant, cf. (2.11), we may replace the infinite lattice $\mathbb{Z}^q$ with a finite set, which is most naturally given by $\{-N, -N+1, \dots, 0, \dots, N-1, N\}^q$, where $N > 0$ is an integer. Then $m \leq (2N+1)^q$; in Bianchini et al. (2020), $N = 50$ for $q = 1, 2$. Bardenet and Titsias (2015) suggested an alternative approach, which does not require the spectral approach used above but specifies the DPP density directly by (2.10) and exploits certain bounds for the product in (2.11). However, it is then harder to interpret the parameters, and in particular to work with an intensity parameter.

To fix the values of α and β, we could follow Lavancier et al. (2015) who proposed to approximate some summaries such as the pair correlation function which depends only on (α, β). Instead, in Section 2.7, we discuss an empirical Bayesian approach to select hyperparameters and hyperpriors for both the Strauss process given by (2.3)-(2.5) and the DPP given by (2.12)-(2.13).

2.4 Normalized infinite divisible prior for the weights

When deriving full conditional distributions for our Metropolis-within-Gibbs sampler given in Section 2.6, it becomes convenient to introduce ancillary variables t and u as specified below.

Conditioned on m, let $\boldsymbol{s} = (s_1, \ldots, s_m)$ consists of iid positive continuous random variables, with the distribution of each s_h not depending on m, and with $\boldsymbol{s}$ independent of $(\xi, \boldsymbol{\mu}, \boldsymbol{\gamma})$. Set $t = \sum_{h=1}^{m} s_h$ and $\boldsymbol{w} = (s_1/t, \ldots, s_m/t)$, so $\boldsymbol{s}$ and $(\boldsymbol{w}, t)$ are in a one-to-one correspondence. In particular, in Sections 2.8-2.9, we assume that each s_h follows a gamma distribution, in which case our model can be referred to as a *finite Dirichlet mixture model with repulsive locations.* We point out that the idea of building the weights $\boldsymbol{w}$ by normalization not only has computational advantages – as previously discussed – but it also allows us to embed the model into the large class of mixtures obtained by normalization of finite point processes (Argiento and De Iorio, 2022). This latter class, to be defined, requires only the distribution of s_h's to be infinitely divisible, and it is the finite-dimensional counterpart of the normalized random measures with independent increments, which has been thoroughly investigated in the last two decades in the Bayesian nonparametric literature (see, for instance, Regazzini et al., 2003; James et al., 2009; Lijoi and Prünster, 2010). The weights $\boldsymbol{w}$ resulting from a finite normalization have distribution on the simplex that is denominated as *normalized infinite divisible* following Favaro et al. (2011). See also Lijoi et al. (2020a). It is worth underlining that our Metropolis-within-Gibbs sampler, cf. Section 2.6, works also for normalized infinite divisible priors different from the Dirichlet distribution, such as those introduced in Argiento and De Iorio (2022).

One of the advantages of building the distribution of the weights $\boldsymbol{w}$ by normalization is that computations are easier. The main idea is to consider a gamma random variable v with scale parameter one and shape parameter n, which is independent of $(\xi, \boldsymbol{\mu}, \boldsymbol{\gamma}, \boldsymbol{s}, c)$. Then, we set the ancillary variable $u := v/t$. It is immediate to show that u is well defined, i.e., it has a density with respect to the Lebesgue measure given by

$$p(u) = \frac{u^{n-1}}{\Gamma(n)} \int_0^{\infty} t^n \mathrm{e}^{-ut} p(t) dt$$

where $p(\cdot)$ in the integral is the density function of t. We show below (see (2.18)) that, conditioned on u, the full conditional of the unnormalized weights s_h's factorize (i.e., the weights are conditionally independent), so that simulation will be drastically simplified. We notice that the *trick* of introducing the ancillary variable u is familiar in the context of normalized completely random measure as mixing measures for mixture models. It was studied in James et al. (2009) in the infinite dimensional case and largely exploited by Argiento and De Iorio (2022) and Argiento et al. (2016) in the finite dimensional setting.

2.5 Posterior distribution and a useful decomposition of the cluster centers

To specify the posterior obtained by considering all parameters introduced so far, including $(\boldsymbol{s}, t, u)$, we first notice that the dominating measure implicitly used in (2.2) leads to a new dominating measure ν given as follows. Let Ξ and Γ denote the spaces where ξ and each γ_h take values, respectively, equipped with some appropriate σ-algebras and measures $\mathrm{d}\xi$ and $\mathrm{d}\gamma_h$ (typically Borel σ-algebras and Lebesgue measures). For $m = 1, 2, \ldots$, set $\boldsymbol{s}_m = (s_1, \ldots, s_m)$ and $\boldsymbol{\gamma}_m = (\gamma_1, \ldots, \gamma_m)$, let $\mathrm{d}\boldsymbol{s}_m$ denote the Lebesgue measure on $\mathbb{R}_+^m$, let $\mathrm{d}\boldsymbol{\gamma}_m$ denote the product measure $\prod_{h=1}^{m} \mathrm{d}\gamma_h$, and consider arbitrary measurable subsets $A \subseteq \Xi$, $B_m \subseteq \Omega_m$, $C_m \subseteq \mathbb{R}_+^m$, $D_m \subseteq \Gamma^m$, $E_m \subseteq \{1, \ldots, m\}^n$, and $F \subseteq \mathbb{R}_+$ (we still let the σ-algebra of Ω_m be induced by the Borel σ-algebra of $\mathbb{R}^{qm}$ and the mapping $\mathbb{R}^{qm} \ni (\mu_1, \ldots, \mu_m) \mapsto \{\mu_1, \ldots, \mu_m\} \in \Omega_m$ with $\mu_1, \ldots, \mu_m$ pairwise distinct). Then the measure $\mathrm{d}\boldsymbol{\mu}$ together with the other reference measures leads to

$$\begin{aligned}&\nu(A \times \{\cup_{m=0}^{\infty} B_m \times C_m \times D_m \times E_m\} \times F) \\ &\quad = \int_A \mathrm{d}\xi \sum_{m=0}^{\infty} \frac{1}{m!} \int_{B_m} \mathrm{d}\boldsymbol{\mu}_m \int_{C_m} \mathrm{d}\boldsymbol{s}_m \int_{D_m} \mathrm{d}\boldsymbol{\gamma}_m \sum_{c_1,\dots,c_n=1}^{m} \mathbb{I}[\boldsymbol{c} \in E_m] \int_F \mathrm{d}u. \end{aligned} \tag{2.14}$$

The posterior density of the new parameter $(\xi, \boldsymbol{\mu}, \boldsymbol{s}, \boldsymbol{\gamma}, \boldsymbol{c}, u)$ with respect to ν is then

$$\begin{aligned}&p(\xi, \boldsymbol{\mu}, \boldsymbol{s}, \boldsymbol{\gamma}, \boldsymbol{c}, u \,|\, \boldsymbol{y}) \propto p(\xi)p(\boldsymbol{\mu} \,|\, \xi) \times \\ &\qquad \left[\prod_{h=1}^{m} p(\gamma_h)p(s_h)\right] p(u \,|\, t)\frac{1}{t^n} \left[\prod_{i=1}^{n} s_{c_i} f(y_i \,|\, (\mu_{c_i}, \gamma_{c_i}))\right]. \end{aligned} \tag{2.15}$$

In the algorithm for posterior simulation presented in Section 2.6, we find it useful to split $\boldsymbol{\mu}$ into those cluster centres which are used to allocate the data, and those which are not, that is, $\boldsymbol{\mu}^{(a)} = \{\mu_{c_1}, \dots, \mu_{c_n}\}$ and $\boldsymbol{\mu}^{(na)} = \boldsymbol{\mu} \setminus \boldsymbol{\mu}^{(a)}$. For the points of these processes we use the notation $\boldsymbol{\mu}^{(a)} = \{\mu_1^{(a)}, \dots, \mu_k^{(a)}\} =$ and $\boldsymbol{\mu}^{(na)} = \{\mu_1^{(na)}, \dots, \mu_\ell^{(na)}\}$. Note that $1 \le k \le m$, $\ell \ge 0$, and the product measure $\mathrm{d}\boldsymbol{\mu} \times \mathrm{d}\boldsymbol{\mu}$ on $\Omega \times \Omega$ lifted by the map $(\boldsymbol{x}, \boldsymbol{z}) \mapsto \boldsymbol{x} \cup \boldsymbol{z}$ results in the measure $\mathrm{d}\boldsymbol{\mu}$. Hence, $(\boldsymbol{\mu}^{(a)}, \boldsymbol{\mu}^{(na)})$ conditioned on ξ has density

$$p(\boldsymbol{\mu}^{(a)}, \boldsymbol{\mu}^{(na)} \,|\, \xi) = p(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \,|\, \xi)$$

with respect to the product measure $\mathrm{d}\boldsymbol{\mu}^{(a)} \times \mathrm{d}\boldsymbol{\mu}^{(na)}$ (thinking of the measures $\mathrm{d}\boldsymbol{\mu}, \mathrm{d}\boldsymbol{\mu}^{(a)}, \mathrm{d}\boldsymbol{\mu}^{(na)}$ as being identical but of course not thinking of $\boldsymbol{\mu}, \boldsymbol{\mu}^{(a)}, \boldsymbol{\mu}^{(na)}$ as being identical).

Obviously, $(\boldsymbol{\mu}, \boldsymbol{s}, \boldsymbol{\gamma}, \boldsymbol{c})$ and $(\boldsymbol{\mu}^{(a)}, \boldsymbol{s}^{(a)}, \boldsymbol{\gamma}^{(a)}, \boldsymbol{\mu}^{(na)}, \boldsymbol{s}^{(na)}, \boldsymbol{\gamma}^{(na)}, \boldsymbol{c})$ are in a one-to-one correspondence, and the cardinalities of the point processes $\boldsymbol{\mu}^{(a)}$ and $\boldsymbol{\mu}^{(na)}$ satisfy $1 \le k < +\infty$ and $0 \le \ell < +\infty$. Finally, setting $n_h = \#\{i : c_i = h\}$, we obtain from (2.14) and (2.15) the posterior density

$$\begin{aligned}&p(\xi, \boldsymbol{\mu}^{(a)}, \boldsymbol{s}^{(a)}, \boldsymbol{\gamma}^{(a)}, \boldsymbol{c}, \boldsymbol{\mu}^{(na)}, \boldsymbol{s}^{(na)}, \boldsymbol{\gamma}^{(na)}, u \,|\, \boldsymbol{y}) \propto \\ &\quad p(\xi)p(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \,|\, \xi) \left[\prod_{h=1}^{k} p(\gamma_h^{(a)})p(s_h^{(a)})(s_h^{(a)})^{n_h} \prod_{i:c_i=h} f(y_i \,|\, (\mu_h^{(a)}, \gamma_h^{(a)}))\right] \\ &\qquad \times \left[\prod_{h=1}^{\ell} p(\gamma_h^{(na)})p(s_h^{(na)})\right] p(u \,|\, t)\frac{1}{t^n} \end{aligned} \tag{2.16}$$

with respect to a new dominating measure defined by (using an obvious notation)

$$\begin{aligned}&\nu'(A \times \left\{\cup_{k=1}^{\infty} B_k^{(a)} \times C_k^{(a)} \times D_k^{(a)} \times E_k^{(a)}\right\} \times \left\{\cup_{\ell=0}^{\infty} B_\ell^{(na)} \times C_\ell^{(na)} \times D_\ell^{(na)}\right\} \times F) \\ &\quad = \int_A \mathrm{d}\xi \sum_{k=1}^{\infty} \frac{1}{k!} \int_{B_k^{(a)}} \mathrm{d}\boldsymbol{\mu}_k^{(a)} \int_{C_k^{(a)}} \mathrm{d}\boldsymbol{s}_k^{(a)} \int_{D_k^{(a)}} \mathrm{d}\boldsymbol{\gamma}_k^{(a)} \\ &\qquad \sum_{\substack{c_1,\dots,c_n=1: \\ \#\{c_1,\dots,c_n\}=k}}^{k} \mathbb{I}[\boldsymbol{c} \in E_k^{(a)}] \times \sum_{\ell=0}^{\infty} \frac{1}{\ell!} \int_{B_\ell^{(na)}} \mathrm{d}\boldsymbol{\mu}_\ell^{(na)} \int_{C_\ell^{(na)}} \mathrm{d}\boldsymbol{s}_\ell^{(na)} \int_{D_\ell^{(na)}} \mathrm{d}\boldsymbol{\gamma}_\ell^{(na)} \int_F \mathrm{d}u. \end{aligned} \tag{2.17}$$

Without introducing the ancillary variable u, that is, when leaving out the term $p(u \,|\, t)$ in (2.16)), it becomes difficult to derive the full conditionals for the allocated and non-allocated variables $\boldsymbol{\mu}^{(a)}, \boldsymbol{s}^{(a)}, \boldsymbol{\gamma}^{(a)}, \boldsymbol{\mu}^{(na)}, \boldsymbol{s}^{(na)}, \boldsymbol{\gamma}^{(na)}$. This is due to the term $1/t^n$ in (2.16), noting that $t = \sum_{h=1}^{k} s_h^{(a)} + \sum_{h=1}^{\ell} s_h^{(na)}$, which makes it impossible to factorize according to the allocated and non-allocated variables. When including u, we obtain

$$p(u \mid t)\frac{1}{t^n} = \frac{u^{n-1}}{(n-1)!}\exp(-ut)t^n\frac{1}{t^n} =$$
$$\frac{u^{n-1}}{(n-1)!}\left[\prod_{h=1}^{k}\exp(-us_h^{(a)})\right]\left[\prod_{h=1}^{\ell}\exp(-us_h^{(na)})\right], \quad (2.18)$$

which does not depend on t. Using (2.16) and (2.18), a factorization is obtained, which is useful for the Metropolis-within-Gibbs sampler described in the following section.

2.6 Algorithm for posterior simulation

2.6.1 Metropolis-within-Gibbs sampler

In our Metropolis-within-Gibbs sampler for simulating from the posterior (2.16), a single iteration is given by updating from full conditionals for five blocks of variables as specified by the first line in the following steps (A)-(E). Note that we use the notation $p(\cdot|\cdots)$ to indicate that we consider a variable or collection of variables $\cdot$ given the remaining variables $\cdots$ (including the data $\boldsymbol{y}$).

(A) Update the non-allocated variables $(\boldsymbol{\mu}^{(na)}, \boldsymbol{s}^{(na)}, \boldsymbol{\gamma}^{(na)})$ from their full conditional as given by the following steps (i)-(iii), noting the following. Since the cardinality of each of the vectors $\boldsymbol{s}^{(na)}$ and $\boldsymbol{\gamma}^{(na)}$ agrees with the cardinality of $\boldsymbol{\mu}^{(na)}$, it is of paramount importance to resort to a *collapsed* Gibbs sampler. Therefore, in (i) we sample $\boldsymbol{\mu}^{(na)}$ from the conditional density obtained by integrating out $(\boldsymbol{s}^{(na)}, \boldsymbol{\gamma}^{(na)})$, and then in (ii)-(iii) we sample $\boldsymbol{s}^{(na)}$ and $\boldsymbol{\gamma}^{(na)}$ from their respective full conditionals, hence knowing the cardinality ℓ of $\boldsymbol{\mu}^{(na)}$.

(i) Sample from the conditional density obtained by integrating out $(\boldsymbol{s}^{(na)}, \boldsymbol{\gamma}^{(na)})$ and given by

$$p(\boldsymbol{\mu}^{(na)} \mid \xi, \boldsymbol{\mu}^{(a)}, \boldsymbol{s}^{(a)}, \boldsymbol{\gamma}^{(a)}, \boldsymbol{c}, u, \boldsymbol{y}) \propto p(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \mid \xi)\psi(u)^{\ell} \quad (2.19)$$

with respect to $\mathrm{d}\boldsymbol{\mu}^{(na)}$. Here, $\psi(u)$ denotes the Laplace transform of the density $p(s_h)$, and we can got rid of the last term $\psi(u)^{\ell}$, since ℓ is the cardinality of $\boldsymbol{\mu}^{(na)}$. In Section 2.6.2 we verify (2.19) and give details for simulation from (2.19). If, after this update, $\ell = 0$, the following two items (ii) and (iii) are skipped.

(ii) Sample $\boldsymbol{s}^{(na)}$ from its full conditional,

$$p(\boldsymbol{s}^{(na)} \mid \cdots) \propto \prod_{h=1}^{\ell} p(s_h^{(na)})\exp(-us_h^{(na)}).$$

That is, sample independently ℓ values from the exponential tilting of the prior density. Depending on the specific choice of $p(s_h^{(na)})$, this can be done exactly or requires a Metropolis-Hastings step.

(iii) Sample $\boldsymbol{\gamma}^{(na)}$ from its full conditional,

$$p(\boldsymbol{\gamma}^{(na)} \mid \cdots) \propto \prod_{h=1}^{\ell} p(\gamma_h^{(na)}).$$

That is, sample independently ℓ values from the prior density $p(\gamma_h^{(na)})$.

(B) Update the allocated variables $(\boldsymbol{\mu}^{(a)}, \boldsymbol{s}^{(a)}, \boldsymbol{\gamma}^{(na)})$:

(i) Sample $\boldsymbol{\mu}^{(a)}$ from its full conditional,

$$p(\boldsymbol{\mu}^{(a)} \mid \cdots) \propto p(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \mid \xi) \prod_{h=1}^{k} \left[\prod_{i:c_i=h} f(y_i \mid (\mu_h^{(a)}, \gamma_h^{(a)})) \right],$$

where by (2.6) we can replace $p(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \mid \xi)$ by $g(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \mid \xi)$. We do this by updating each of $\mu_h^{(a)}$ from

$$p(\mu_h^{(a)} \mid \cdots) \propto g(\mu_h^{(a)} \cup \{\boldsymbol{\mu}^{(a)} \setminus \{\mu_h^{(a)}\}\} \cup \boldsymbol{\mu}^{(na)}) \prod_{i:c_i=h} f(y_i \mid (\mu_h^{(a)}, \gamma_h^{(a)})).$$

Appendix Section 2.A.1 discusses how to construct a proposal density for sampling from $p(\mu_h^{(a)} \mid \cdots)$ using a Metropolis-Hastings step.

(ii) Sample $\boldsymbol{s}^{(a)}$ from its full conditional,

$$p(\boldsymbol{s}^{(a)} \mid \cdots) \propto \prod_{h=1}^{k} (s_h^{(a)})^{n_h} e^{-us_h^{(a)}} p(s_h^{(a)}).$$

Here, the $s_h^{(a)}$'s are independent conditional to everything else, so they can be updated individually using a Metropolis-Hastings step.

(iii) Sample $\boldsymbol{\gamma}^{(a)}$ from its full conditional,

$$p(\boldsymbol{\gamma}^{(a)} \mid \cdots) \propto \prod_{h=1}^{k} p(\gamma_h^{(a)}) \prod_{i:c_i=h} f(y_i \mid (\mu_h^{(a)}, \gamma_h^{(a)})).$$

Unless $p(\gamma_h^{(a)})$ and $f(y_i \mid (\mu_h^{(a)}, \gamma_h^{(a)}))$ are conjugate, we apply a Metropolis step for the $\gamma_h^{(a)}$'s.

Since in this step we have conditioned with respect to $\boldsymbol{c}$ too, k denotes the number of clusters and is fixed.

(C) Sample each c_i from its full conditional, which is a discrete distribution over $1, \ldots, k + \ell$ given by

$$\begin{aligned} p(c_i = h \mid \cdots) &\propto s_h^{(a)} f(y_i \mid (\mu_h^{(a)}, \gamma_h^{(a)})), \qquad h = 1, \ldots, k, \\ p(c_i = k + h \mid \cdots) &\propto s_h^{(na)} f(y_i \mid (\mu_h^{(na)}, \gamma_h^{(na)})), \qquad h = 1, \ldots, \ell. \end{aligned}$$

After this, with a positive probability it may happen that $c_i > k$ for some i's, so that some non-allocated components have become allocated, and some allocated components have become non-allocated. Then a simple relabelling of $(\boldsymbol{\mu}^{(a)}, \boldsymbol{s}^{(a)}, \boldsymbol{\gamma}^{(a)}, \boldsymbol{\mu}^{(na)}, \boldsymbol{s}^{(na)}, \boldsymbol{\gamma}^{(na)})$ and $\boldsymbol{c}$ is needed, so that $\boldsymbol{c}$ takes values in $\{1, \ldots, k\}^n$.

(D) Sample ξ from its full conditional,

$$p(\xi \mid \cdots) \propto p(\xi) p(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \mid \xi).$$

This requires a Metropolis-Hastings step, which is not straightforward when Z_ξ in (2.6) is not expressible in closed form, e.g. in the case of a repulsive pairwise interaction point process. Details on how this issue is overcome are given in Section 2.6.3.

(E) Sample u from its full conditional, which is just a gamma distribution with shape parameter n and inverse scale t.

2.6.2 Updating the non-allocated variables

This section provides the remaining details of step (A)(i). By (2.16),

$$
\begin{aligned}
p(\boldsymbol{\mu}^{(na)} \mid \xi, &\boldsymbol{\mu}^{(a)}, \boldsymbol{S}^{(a)}, \boldsymbol{\gamma}^{(a)}, \boldsymbol{c}, u, \boldsymbol{y}) \\
&= \int\int p(\boldsymbol{\mu}^{(na)}, \boldsymbol{s}^{(na)}, \boldsymbol{\gamma}^{(na)} \mid \xi, \boldsymbol{\mu}^{(a)}, \boldsymbol{S}^{(a)}, \boldsymbol{\gamma}^{(a)}, \boldsymbol{c}, u, \boldsymbol{y}) \, \mathrm{d}\boldsymbol{s}^{(na)} \, \mathrm{d}\boldsymbol{\gamma}^{(na)} \\
&\propto \int\int p(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \mid \xi) \left[\prod_{h=1}^{\ell} p(\gamma_h^{(na)}) p(s_h^{(na)})\right] p(u \mid t) \frac{1}{t^n} \, \mathrm{d}\boldsymbol{s}^{(na)} \, \mathrm{d}\boldsymbol{\gamma}^{(na)} \\
&\propto \int p(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \mid \xi) \left[\prod_{h=1}^{\ell} \exp(-u s_h^{(na)}) p(s_h^{(na)})\right] \mathrm{d}\boldsymbol{s}^{(na)} \qquad (2.20) \\
&= p(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \mid \xi) \psi(u)^{\ell} \qquad (2.21)
\end{aligned}
$$

where (2.20) follows by integrating over $\gamma_h^{(na)}$ and using (2.18), and (2.21) by applying the definition of $\psi(u)$. This verifies (2.19).

Note that (2.19) identifies an unnormalized density for $\boldsymbol{\mu}^{(na)}$ with respect to $\mathrm{d}\boldsymbol{\mu}^{(na)}$. In our examples, the unnormalized density in (2.21) will be hereditary, that is, $p(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \mid \xi) > 0$ implies $p(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}'^{(na)} \mid \xi) > 0$ whenever $\boldsymbol{\mu}'^{(na)}$ consists of one more point than $\boldsymbol{\mu}^{(na)}$. Moreover, in our examples, this density is defined on a compact set, and so we can easily employ the birth-death Metropolis-Hastings algorithm in Geyer and Møller (1994). Specifically, we use Algorithm 11.3 in Møller and Waagepetersen (2004).

2.6.3 Sampling the hyperparameter

When the density $p(\boldsymbol{\mu} \mid \xi)$ is expressible in close form, a standard Metropolis-Hastings move can be employed to update ξ from its full conditional. However, when Z_ξ is intractable, it is a doubly-intractable problem, since a ratio of unknown normalizing constants appears in the Hastings ratio. In fact, if $p(\xi'; \xi \mid \cdots)$ is a proposal density for the Metropolis-Hastings step for the full conditional of ξ, then the acceptance ratio amounts to

$$
\alpha(\xi'; \xi \mid \cdots) = \frac{p(\xi') g(\boldsymbol{\mu} \mid \xi') p(\xi; \xi')}{p(\xi) g(\boldsymbol{\mu} \mid \xi) p(\xi'; \xi)} \frac{Z_\xi}{Z_{\xi'}}, \qquad (2.22)
$$

which is intractable due to the term $Z_\xi / Z_{\xi'}$. To overcome this issue, we can use the exchange algorithm described in Murray et al. (2006) and inspired by the *single auxiliary variable method* proposed by Møller et al. (2006). For further details, see Appendix 2.A.2. This algorithm requires generating an ancillary variable following the same distribution of $\boldsymbol{\mu}$ given ξ'. To this end, we employ the dominated coupling from the past algorithm in Kendall and Møller (2000).

In previous literature, the use of the ancillary variable algorithms in Møller et al. (2006) and Murray et al. (2006) has been limited because of the high computational cost associated to perfect simulation. In contrast, in our examples perfect simulation is fast. As an example, approximating the density of a DPP with $N = 50$ in dimension $q = 2$, as done in Bianchini et al. (2020), is around 25 times more expensive than running a perfect simulation from a Strauss process with parameters and prior for ξ chosen as in Section 2.7.1; for further comparisons, see Appendix 2.B.2. The perfect simulation step is very fast since m (the number of components in the mixture model) is typically small (in our examples it is always less than 10). However, when dealing with applications with a very large number of clusters, such as topic modeling, where the number of clusters is usually between 50 and 100, cf. Blei et al. (2003), we expect perfect simulation to be potentially a bottleneck and limit the use of the exchange algorithm. Although not investigated here, in such cases we could avoid perfect simulation by replacing the exact exchange algorithm of Murray et al. (2006) with asymptotically exact algorithms that should offer smaller computational cost; see for instance Lyne et al. (2015) and Liang et al. (2016).

2.7 Prior elicitation

In this section, we discuss prior elicitation and how to set hyperparameters when the prior for $\boldsymbol{\mu}$ is the Strauss process or the DPP with power exponential spectral density.

2.7.1 Prior elicitation for the Strauss process

Consider the Strauss process prior given by (2.3)-(2.5). In addition to the parameter ξ which controls the intensity, the process depends on two parameters $\alpha \in [0,1]$ and $\delta > 0$ which control repulsiveness and the range of interaction, respectively. Initially we investigated cases where α and δ were random, but then our simulated datasets yielded a large number of clusters a posteriori. Moreover, when fitting mixtures of Gaussian densities to data generated from heavy-tailed distributions, as also discussed at the beginning of Section 2.8, in general better density estimates were obtained when using a larger (i.e., larger than the true value) number of components in the mixture model. For this reason, we obtained a posteriori values of α and δ that induced less repulsive behaviors than desired. Therefore, we suggest below to fix α and δ via an empirical Bayes procedure, and let only ξ be random.

We propose to estimate α and δ as follows. Denote by $p(r)$ the kernel density estimate of the empirical distribution of the pairwise distances between observations; in all the examples, we have obtained such an estimate using the default bandwith selection procedure in Python's `scipy` package. Since δ should be large enough to induce repulsion of redundant clusters, but not too large to affect density estimation severely, we suggest to fix δ as the smallest local minimum point of $p(r)$, that is, $\delta = \min_{r>0}\{r : r \text{ is a local minimum for } p(r)\}$. Further, α should be small enough to encourage separation between the allocated means. Consider, for example, the case with two clusters $\{y_i : c_i = h'\}$ and $\{y_i : c_i = h''\}$ where $0 < \|\mu_{h'} - \mu_{h''}\| \leq \delta$ but the distances from $\mu_{h'}$ and $\mu_{h''}$ to all the other μ_h's are greater than δ. Then, by (2.15), the full conditional of $\mu_{h'}$ has density

$$p(\mu_{h'} \mid \cdots) \propto \alpha \prod_{i:\, c_i = h'} f(y_i \mid \mu_{h'}, \gamma_{h'}).$$

Now, the point of using a repulsive prior is that if the cardinality of cluster h', that is $\#\{i : c_i = h'\}$, is small, the repulsiveness should prevail on the within-cluster likelihood: That is, regardless of how well the value of parameter $\mu_{h'}$ 'fits' data in cluster h', the full conditional density associated to that value should be small because $\mu_{h'}$ is near to the cluster center $\mu_{h''}$. A rough estimate of α can be obtained by assuming $\alpha = \exp(-n^* \log(k_s))$. Here, n^* represents the minimum cluster size needed to balance the repulsive behavior induced by the prior, while k_s represents a 'guess' of $f(\cdot \mid \cdot)$ in a small cluster. In our experiments, we assumed that clusters with less than 5% of the data should be considered small and thus we fixed $n^* = n/20$. Further, we fixed $\log(k_s) = 1$ so that this term did not affect the definition of α. In addition, preliminary sensitivity analysis on α led us to conclude that posterior inference is robust.

Finally, we assume that ξ is random. An upper bound for the expected number of points in $\boldsymbol{\mu}$ is $\xi|R|$, and given an upper bound $M_{\max}$ on the expected number of components, we assume the prior for ξ to be uniform over the interval $\big(|R|^{-1}, M_{\max}|R|^{-1}\big)$. Since the number of clusters is smaller than the number of components, $M_{\max}$ is an upper bound for both, to be fixed in each application according to prior belief.

2.7.2 Prior elicitation for the power exponential spectral DPP model

For the DPP defined on $\mathbb{R}^q$ by the spectral density χ used in (2.13), existence is ensured if $0 < \alpha \leq \alpha_{\max}$, where

$$(\alpha_{\max})^q = \frac{\pi^{q/2}\Gamma(q/\beta + 1)}{\xi\Gamma(q/2 + 1)},$$

cf. Lavancier et al. (2015). So we let $\alpha = s\ \alpha_{\max}$ with $0 < s < 1$ (as specified below), which implies existence of the DPP restricted to any compact subset of $\mathbb{R}^q$. Note that the DPP density given by (2.12)-(2.13) refers to the case $R = [-1/2, 1/2]^q$, and a simple rescaling is needed in the density expression when we fix R to be the smallest rectangle containing all the observations, cf. Lavancier et al. (2015).

Recall that ξ is the expected number of points in $\boldsymbol{\mu}$. We let a priori ξ be uniformly distributed over $[1, M_{\max}]$, where $M_{\max}$ is fixed (as in the case of the Strauss process, cf. Section 2.7.1). As noted in Lavancier et al. (2015), the parameters (s, β) are harder to interpret via (2.13). In our examples, we fix $s = 0.5$ and perform sensitivity analysis on β, concluding that inference is robust.

2.8 Simulation studies

In this section, we compare the reversible jump algorithm in Bianchini et al. (2020) to our Metropolis-within-Gibbs sampler presented in Section 2.6, and show the advantages of repulsive mixtures over non-repulsive ones. We refer to our Metropolis-within-Gibbs sampler as the 'M-w-G sampler' and the reversible jump algorithm as 'RJ'. In Appendix 2.B, we illustrate the advantages of using a Strauss process over a DPP as prior for $\boldsymbol{\mu}$ and provide further simulations when the dimension q or the number of components m increase. In particular, we conclude that the computational cost of posterior inference under the DPP grows exponentially with data dimension q, whilst the computational cost associated to the Strauss process is almost constant as data dimension increases, and that posterior summaries obtained under the DPP and Strauss process are almost identical. This motivates the use of the Strauss process as a prior for $\boldsymbol{\mu}$.

In this section, we study posterior inference in *misspecified* settings, i.e., when the generating process does not coincide with the model used to fit data; for a formal definition of misspecification, see Kleijn et al. (2006). In misspecified settings, there is a trade-off between the accuracy of the density and number of clusters estimation recovered by the mixture model, cf. Guha et al. (2021). This indicates that more accurate density estimates correspond to overestimated number of clusters and vice-versa. In fact, to recover the shape of non-Gaussian data, several Gaussian components (with similar values of the mean parameters) are needed. We expect that the repulsiveness induced by the prior for $\boldsymbol{\mu}$ favours cluster over density estimation.

We consider two simulation scenarios, the first one is as in Miller and Dunson (2019), where the authors generated iid data $y_1, \dots, y_n$ using a two-step procedure as follows. First, a mixture density f_0 with m_0 components is selected. Second, a random density $\widetilde{f}$ is drawn from a Dirichlet process mixture, with base measure given by f_0. Specifically,

$$\begin{aligned} y_1, \dots, y_n \mid P &\overset{\text{iid}}{\sim} \widetilde{f}(\cdot) = \int \mathcal{N}(\cdot \mid \theta, 0.25^2) P(\mathrm{d}\theta), \\ P \sim DP(af_0), &\qquad f_0 = \sum_{h=1}^{m_0} w_{0h} \mathcal{N}(\mu_{0h}, \sigma_{0h}^2), \end{aligned} \tag{2.23}$$

where $DP(af_0)$ denotes the Dirichlet process with total mass parameter a and centering probability measure induced by f_0. We fix $a = 500$, $m_0 = 4$, $\boldsymbol{w}_0 = (0.25, 0.25, 0.3, 0.2)$, $\boldsymbol{\mu}_0 = (-3.5, 3, 0, 6)$, and $\boldsymbol{\sigma}_0 = (0.8, 0.5, 0.4, 0.5)$. Following Miller and Dunson (2019), we

Params.		RJ		M-w-G sampler		
ξ	β	ESS	$\mathbb{E}[m \mid \text{data}]$	ESS	$\mathbb{E}[m \mid \text{data}]$	$\mathbb{E}[k \mid \text{data}]$
4	10	90.63	4.33	8201.41	4.01	4.00
4	2.5	62.46	4.402	3735.80	4.01	4.00
4	25	83.32	4.44	2971.05	4.02	4.00

Table 2.1: Summary of the MCMC simulations for the reversible jump algorithm (RJ) in Bianchini et al. (2020) and our Metropolis-within-Gibbs sampler (M-w-G sampler). ESS denotes the effective sample size out of the 10,000 MCMC samples.

interpret the data generating density $\widetilde{f}$ as a perturbation of the 'true' density f_0, so the goal is to recover f_0 and m_0.

The second simulation scenario considers draws from the following mixture of two components:

$$y_1, \dots y_n \overset{\text{iid}}{\sim} 0.5\, t_q(1, \boldsymbol{\mu}_0, \Sigma_0) + 0.5\, MSN_q(\omega, \mu_1, \sigma_1). \tag{2.24}$$

Here $t_q(1, \boldsymbol{\mu}_0, \Sigma_0)$ denotes the density of a q-dimensional Student distribution with one degree of freedom, location $\boldsymbol{\mu}_0$, and scale matrix Σ_0. Furthermore, $MSN_d(\omega, \mu_1, \sigma_1)$ denotes the density of a q-dimensional random vector, where each entry is drawn independently from a skew normal distribution with mean $\mu_1 + \omega\sigma_1\sqrt{2/\pi}$, being μ_1 the location parameter and ω the scale parameter of the skew normal distribution. The dimension q and the other parameters in (2.24) will be specified later.

For both scenarios, the kernel $f(\cdot \mid \cdot)$ in (2.1) In addition to the prior assumptions (a)-(e) in Section 2.1, we let a priori $(w_1, \dots, w_m) = (s_1/t, \dots, s_m/t)$, with $\boldsymbol{s}$ as in Section 2.4, where each s_h follows a gamma distribution with shape and scale equal to one, Finally, unless otherwise stated, parameters of the Strauss point process or the DPP are chosen as discussed in Sections 2.7.1-2.7.2. In particular, we fix $M_{\max} = 30$.

2.8.1 Monitoring MCMC mixing

In this section, data are given by 500 observations simulated in accordance to (2.23). The marginal prior for $\boldsymbol{\mu}$ is the DPP specified in Section 2.7.2, where in order to identify the effect of the algorithm on posterior inference, we keep the intensity parameter ξ fixed. Furthermore, we consider three possible values for the hyperparameters ξ and β in the DPP prior, cf. Table 2.1. For each choice of hyperparameter values, we ran both MCMC algorithms (M-w-G sampler and RJ) for 20,000 iterations, discarding the first 10,000 as burn-in and without thinning the chain. In order to compare the results, we consider the effective sample size (ESS) of the number of components in the mixture (m in our notation) as well as its autocorrelation.

Table 2.1 reports, for different combination of hyperparameters, the posterior expected value of m as well as the effective sample sizes for m obtained by the two algorithms. Since in our M-w-G sampler the number of clusters can be smaller than m, the table also shows the posterior expected value of k (the number of allocated components/clusters). Figure 2.1 shows for both algorithms trace plots and autocorrelation plots for m when $\xi = 4$ and $\beta = 10$ (first row of Table 2.1). Note that both algorithms offer good estimates of the number of components in the mixture. However, the performance of our M-w-G sampler is superior to the RJ algorithm in all the settings of hyperparameters we tested: Our M-w-G sampler generally produces a (much) higher effective sample size and overall better mixing of the chains.

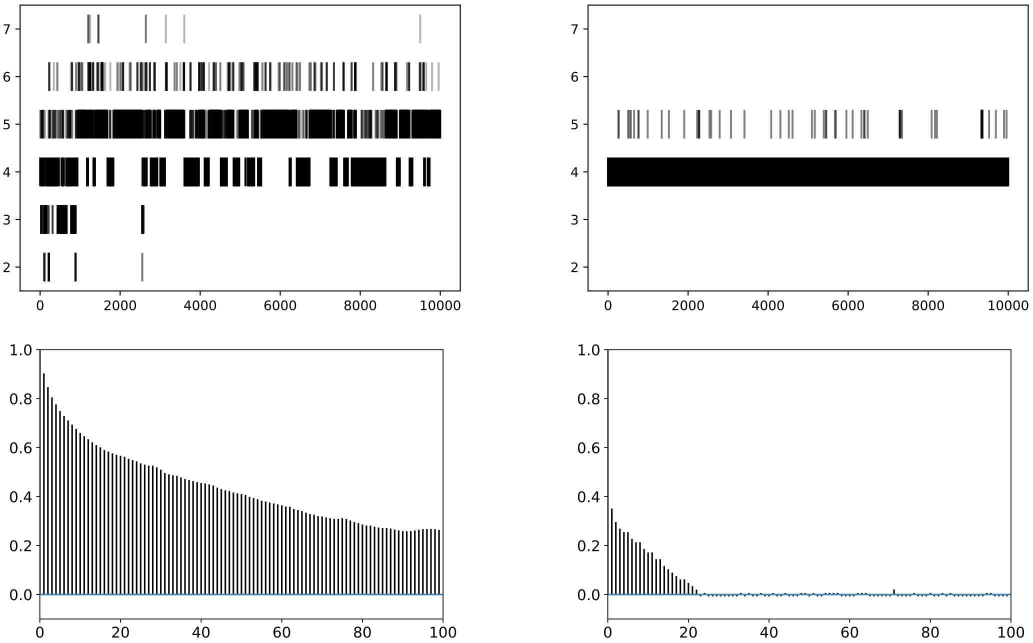


Figure 2.1: Trace plots (top) and autocorrelations (bottom) of m when $\xi = 4$ and $\beta = 10$. Left: RJ. Right: M-w-G sampler.

2.8.2 Comparison with DPM and FM

We focus on the differences between repulsive and non-repulsive mixtures using two further simulations. For the class of non-repulsive mixtures, we consider (i) the finite mixture models (FM) of Gaussian densities in Argiento and De Iorio (2022) and Miller and Harrison (2018), and (ii) the Dirichlet Process Mixture (DPM) of Gaussian densities.

Both FM and DPM require the choice of a base measure P_0 that we fix as the normal-inverse-Wishart distribution (or the normal-inverse-gamma distribution in the univariate case). Hyperparameters are fixed according to Fraley and Raftery (2007) to provide a weakly informative prior. Moreover, the concentration parameter in the Dirichlet process is fixed to one, and for the FM model we consider as prior for m the shifted Poisson distribution (with support $\{1, 2, \ldots\}$) so that the prior mean of the components is equal to four if the data generating process is (2.23) and to two if the data generating process is (2.24). Finally, for our model, we assume the Strauss process prior for $\boldsymbol{\mu}$.

Posterior simulation from the FM model was carried out using the R package `AntMAN`[1], while for the DPM we used the R package `BNPMix` (Corradin et al., 2020). For all three models, we ran the MCMC algorithm for $100,000$ iterations, discarding the first $50,000$ as a burn-in and keeping one of every ten iterations, so that in each case the final sample size is $5,000$.

In the first simulation study, data are given by 400 simulated observations from (2.23). Figure 2.2 shows the true data generating density, together with Bayesian mixture density estimates obtained by our model and the DPM (left), as well as the distribution of the number of clusters under the three models (right). Here, by Bayesian density estimate we always mean the posterior expectation of the mixture density evaluated on a fixed grid of points. As expected, under this misspecified setting, the use of non-repulsive mixture models overestimate the number of clusters. For instance, to recover the shape of the

[1] available at https://github.com/bbodin/AntMAN

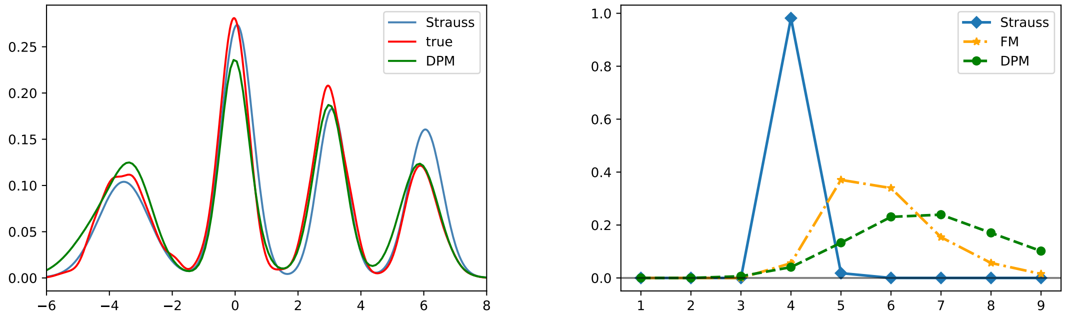


Figure 2.2: Posterior inference based on data simulated from (2.23). To the left, Bayesian mixture density estimates under the Strauss process and the DPM priors for $\boldsymbol{\mu}$, together with the true mixture density which has four components. To the right, posterior distributions of the number of allocated components under the Strauss process, FM, and DPM priors for $\boldsymbol{\mu}$.

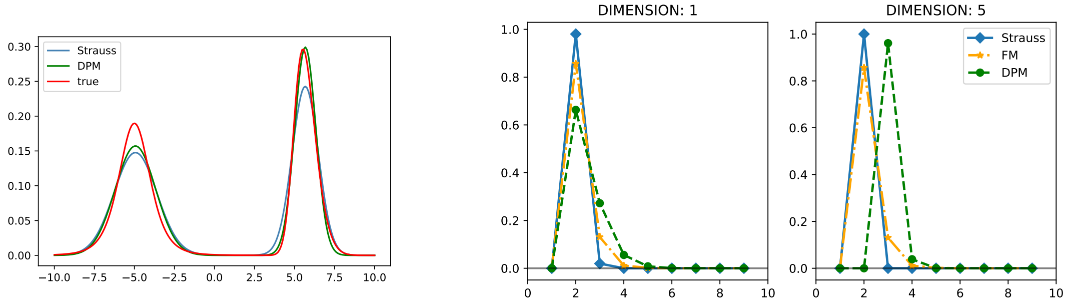


Figure 2.3: Posterior inference based on data simulated from (2.24). To the left, when $q = 1$, Bayesian mixture density estimates under the Strauss process and the DPM priors for $\boldsymbol{\mu}$, together with the true mixture density which has two components. To the right, when $q = 1, 5$, posterior distributions of the number of allocated components under the Strauss process, FM, and DPM priors for $\boldsymbol{\mu}$.

leftmost *bell* of the data generating density in Figure 2.2, several Gaussian components (with close cluster centers) are needed. Our model instead, due to the repulsiveness induced by the prior on $\boldsymbol{\mu}$, 'correctly' identifies four clusters.

For the second simulation study, we simulated 500 observations from (2.24) in each of the cases $q = 1$ and $q = 5$, where we fixed $\mu_0 = (-5, \ldots, -5)$, $\Sigma_0 = I_q$, $\omega = 2$, $\mu_1 = 5$, and $\sigma_1 = 1$. Figure 2.3 reports density estimates when $q = 1$ (left) and the posterior distribution of the number of clusters for the three models (right) when $q = 1, 5$. Note that, among the three models, our is the one that gives highest posterior probability to the true value $k = 2$. When $q = 5$, DPM assigns the highest probability to three clusters. Appendix 2.B.3 contains a comparison of the cluster estimates under the three models considered when $q = 1$, and we conclude that the repulsive mixture model is the one that better recovers the true clustering of the data in this example.

2.9 Teenager problematic behavior dataset

In this section, we apply our model, with the Strauss process prior for $\boldsymbol{\mu}$, to a dataset consisting of $n = 6504$ observations coming from the Wave 1 data of the National Longitudinal Study of Adolescent to Adult Health, which is available at http://www.icpsr.umich.edu/icpsrweb/ICPSR/studies/21600. The data were also considered in Collins and Lanza (2009) and Li et al. (2018).

The dataset corresponds to six survey items pertaining problematic behaviors in teenagers, so that for the i'th teenager, $y_i = (y_{i1}, \dots, y_{i6}) \in \{0,1\}^6$ is a binary vector, where $y_{ij} = 1$ means a positive answer to entry j. The six entries correspond to (i) 'lied to parents', (ii) 'loud/rowdy/unruly in a public place', (iii) 'damaged property', (iv) 'stolen from a store', (v) 'stolen something worth less than 50 dollars', and (vi) 'taken part in a group fight'.

We let the kernel in (2.1) be given by

$$f(y \,|\, \mu_h) = \prod_{j=1}^{6} \mu_{hj}^{y_j} \, (1 - \mu_{hj})^{1-y_j}, \quad y = (y_1, \dots, y_6) \in \{0,1\}^6, \tag{2.25}$$

so that the six entries in $y_i = (y_{i1}, \dots, y_{i6})$ are conditionally independent binary random variables with success probability vector $\mu_h = (\mu_{h1}, \dots, \mu_{h6})$. Note that there is no parameter γ, and the probability vector $(\mu_1, \dots, \mu_m)$ belongs to $R = [0,1]^6$. The mixture model with kernel (2.25) is known as a *latent class model.*

As the prior for $\boldsymbol{\mu}$, we assume the Strauss process on R with parameters $\delta = 0.4$, $\alpha = \mathrm{e}^{-n^*}$ (with $n^* = 50$), and a uniform prior on ξ with $M_{\max} = 30$, cf. Section 2.7.1. In this context, we may consider the μ_h's as cluster centres/locations, where repulsion among the μ_h's is meant to favor identification of the clusters.

We ran our posterior simulation algorithm for 20,000 iterations, after discarding other 20,000 iterations as burn-in and saving one of every ten iterations. So the final sample is of size $M = 2,000$, and we denote μ_h^j the value of μ_h at iteration $j = 1, \dots, M$. Below we summarize our findings for the cluster centres and compare to what was obtained in Li et al. (2018), where the authors used a finite mixture model with the same kernel (2.25) as ours, but fixed the number of clusters to be equal to four.

We obtained $P(k = 5 \,|\, \text{data}) \approx 1$. As usually done in Bayesian mixture modelling, we computed a point estimate of the latent partition of the data (as given by the unknown c_i's) by selecting, among the partitions visited during the MCMC iterations, the minimum point of the Binder loss function with equal misclassification cost, cf. Binder (1978). Then, we evaluated the weights in each cluster by $\hat{w}_h = \#\hat{C}_h/n$, $h = 1, \dots, 5$, where $\hat{C}_h$ is the estimated index set of data in cluster h. Furthermore, as in Molitor et al. (2010), we estimated the cluster centres by

$$\hat{\mu}_h^{(a)} = \frac{1}{M} \sum_{j=1}^{M} \frac{1}{\#\hat{C}_h} \sum_{i \in \hat{C}_h} \mu_{c_i}^j, \qquad h = 1, \dots, 5.$$

Figure 2.4 shows these estimates, together with the empirical frequencies in each cluster as given by

$$\mu_h^{\text{emp}} = \frac{1}{\#\hat{C}_h} \sum_{i \in \hat{C}_h} y_i, \qquad h = 1, \dots, 5.$$

Note that in Figure 2.4, the estimated clusters are labeled $(1), \dots, (5)$ and ordered by the estimated weights.

The following interpretation of the clusters is consistent with the one given in Li et al. (2018): Figure 2.4 shows that cluster (1) accounts for 59% of the data and groups teenagers with few problematic behaviours, since all estimated and empirical cluster centers in the leftmost panel in Figure 2.4 are small. Further, cluster (2) groups 18% of the subjects and describes minor problematic behaviours (relating to the first and second survey items).

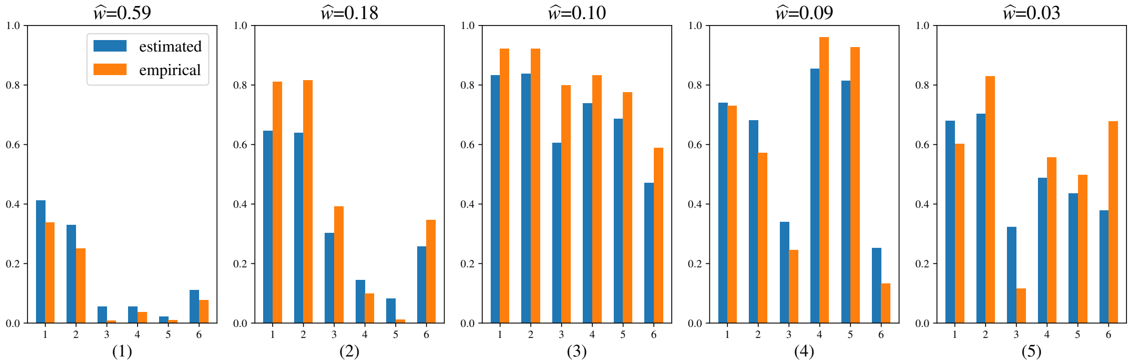


Figure 2.4: Estimated cluster centres $\hat{\mu}_h^{(a)}$ (in blue) and empirical estimates μ_h^{emp} (in orange) when the clusters are sorted according to cluster sizes as given by the estimated weights $\hat{w}_h$, $h = 1, \dots, 5$ (specified at the top of each plot). The clusters are also labelled as $(1), \dots, (5)$ (specified at the bottom of each plot).

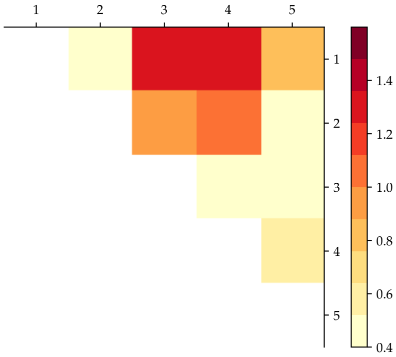


Figure 2.5: Pairwise distances between the estimated $\hat{\mu}_h^{(a)}$, $h = 1, \dots, 5$.

Finally, clusters (3), (4), and (5) represent smaller groups of teenagers who are truly problematic, as their tendency to commit small crimes (cluster 4) or fights (clusters 3 and 5) is very high.

In Figure 2.4, there are discrepancies between the empirical frequencies and our estimates, see for instance the estimates of μ_{h1} and μ_{h2} in cluster (2) and of μ_{h3} and μ_{h6} in cluster (5). These discrepancies can be explained by the use of the repulsive prior, which encourages separation among clusters.

Moreover, Figure 2.5 shows the pairwise Euclidean distances among the estimated $\hat{\mu}_h^{(a)}$, $h = 1, \dots, 5$. Here, the smallest distance is around 0.41, which is close to the value of δ (which we fixed to be equal to 0.4). Note that $\hat{\mu}_5^{(a)}$ is very close to both $\hat{\mu}_2^{(a)}$ and $\hat{\mu}_3^{(a)}$; and $\hat{\mu}_1^{(a)}$ and $\hat{\mu}_2^{(a)}$ are close as well.

Finally, we performed posterior inference with $\delta = 0.5$ and $n^* = 100$ to induce more separation. In this case, our inference gave four estimated clusters, in agreement with Li et al. (2018). However, compared to Figure 2.4, the estimated cluster centres $\hat{\mu}_h^{(a)}$ were then further more different from the empirical frequencies μ_h^{emp}. As noticed in Section 2.8, this trade-off between density versus cluster estimation accuracy is not surprising.

2.10 Discussion

In this work we have contributed to the fast-growing literature on repulsive mixture models. A main contribution is the introduction of a unifying framework which encompasses previously proposed repulsive mixtures as special cases. In our setting, a repulsive point process is assumed as prior for 'cluster centres' of the parametric kernel densities, thus making it more likely having a small number of well separated clusters in the mixture model. In particular, we have showed the usefulness of the Strauss process prior, which is a simple example of a repulsive pairwise interaction point process.

By studying posterior characterization of the repulsive point process, we were able to derive a Metropolis-within-Gibbs sampler that avoids the arduous choice of problem-specific reversible jump proposals (Xu et al., 2016; Bianchini et al., 2020) and the computationally expensive evaluation of infinite summations and integrals over the parameters space (Xie and Xu, 2019) as seen in previous work. When deriving the posterior distribution of the repulsive point process prior, we extended the approach in Argiento and De Iorio (2022) but framing our model within the class of normalized point processes mixture models.

Our MCMC algorithm can also handle cases when the point process density involves an intractable normalizing constant, which has not been considered in the previous literature. In particular, we used an ancillary variable method which eliminates the problem of having a ratio of normalizing constants in the Hastings ratio when making posterior simulations for full conditional of the hyperparameter. Since our mixture model is parsimonious (i.e., the number of components is typically small), the ancillary variable method relying on a perfect simulation algorithm is fast.

We tested our approach by extensive simulation studies, comparing it to the reversible jump approach of Xu et al. (2016) and Bianchini et al. (2020), where we concluded that our Metropolis-within-Gibbs (M-w-G) sampler has better mixing. Our M-w-G sampler scales well with data dimension and this feature was particularly evident when we assumed the Strauss process as a prior for the cluster centers. Furthermore, since repulsive mixture models encourage a small number of well separated components, thus controlling the computational cost, our algorithm was shown to scale well with sample size too.

Finally, we illustrated the advantages of repulsive mixtures against the popular Dirichlet process mixtures and finite mixtures. We concluded that repulsive mixtures are especially useful when the model is misspecified.

Several further extensions are possible. Beyond mixture models for cluster detection, feature allocation problems and regression settings could be considered. Further, adapting our approach to hierarchical and nested settings, where multiple groups of data are present, could be of interest. Finally, extensions of our model to handle extremely high dimension data are also of interest, for instance in the field of genomics, where a repulsive prior would help in deriving interpretable results characterized by few and well separated clusters.

Appendix

2.A Further details on the Metropolis-within-Gibbs sampler used for posterior simulation

This section provides additional details for the Metropolis-within-Gibbs (M-w-G) sampler in Section 2.6.

2.A.1 The choice of the proposal distribution

For most choices of the point process density $p(\boldsymbol{\mu} \mid \xi)$ and the mixture kernel $f(\cdot \mid \cdot)$, the update of the allocated means $\mu_h^{(a)}$ requires sampling from an unnormalized distribution, which we do via a Metropolis-Hastings step. As proposal distribution we use a mixture of two normal distributions with means equal to the current value of $\mu_h^{(a)}$ but with different variances so that

$$p(\mu'; \mu_h^{(a)}) = \kappa \mathcal{N}(\mu' \mid \mu_h^{(a)}, \underline{\sigma}^2 I) + (1-\kappa)\mathcal{N}(\mu' \mid \mu_h^{(a)}, \overline{\sigma}^2 I), \tag{2.26}$$

where $\kappa = 0.9$, $\underline{\sigma} = 0.1$, and $\overline{\sigma} = 1.5$ when $q = 1, 2$ and $\overline{\sigma} = 1.5q$ when $q > 2$. The intuition that led us to consider such a proposal is as follows, where for ease of notation we drop the superscript (a) when considering a current value of $\mu_h^{(a)}$, denoted μ_1, and another cluster centre μ_2. Suppose that μ_1 and μ_2 are close and far from the remaining points in $\boldsymbol{\mu}$. If the number of observations allocated to μ_1 is small, we want a proposal distribution $p(\mu_1'; \mu_1)$ that gives significant mass to values that are far from μ_2, so that, given the repulsiveness of the point process, this proposal is likely to be accepted. This is the case when we sample from the second component of (2.26) (in fact, if μ_1' is far from μ_1, with sufficiently large probability it is far from μ_2 as well). On the other hand, if the number of observations allocated to μ_1 is large, we want a proposal that gives significant mass to a neighborhood of of the current value of μ_1, to get a precise fit of the data. This is what happens if we sample from the first component of (2.26).

For the second component in (2.26), instead of fixing $\overline{\sigma}$ as we do, an alternative is to exploit the properties of $g(\cdot \mid \xi)$ as follows. Suppose we condition on sampling from $\mathcal{N}(\mu_1' \mid \mu_1, \overline{\sigma}^2 I)$ in (2.26). Then $\|\mu_1' - \mu_1\|^2/\overline{\sigma}^2 \sim \chi^2(q)$, the chi-squared distribution with q degrees of freedom. Considering the Strauss density, a possibility is to fix $\overline{\sigma}$ to give sufficiently high mass to values of μ_1' that are outside the range of interaction of μ_1, i.e., such that $P(\|\mu_1' - \mu_1\|^2 > \delta) > p_0$ for some fixed p_0, with the intuition that this gives a positive probability to μ_1' being distant at least δ also from μ_2. Considering the DPP density instead, the same argument holds but replacing δ with the range of correlation r_0, cf. Lavancier et al. (2015). That is, (2.12) implies that C is of the form $C(\mu_1, \mu_2) = C_0(r)$ with $r = \|\mu_1 - \mu_2\|$, and defining the corresponding correlation function $R(r) = C_0(r)/C_0(0)$, r_0 is chosen such that $R(r)$ is effectively zero.

2.A.2 The exchange algorithm and perfect simulation

With the same notation as Section 2.6.3, the exchange algorithm (Murray et al., 2006) consists of the following steps:

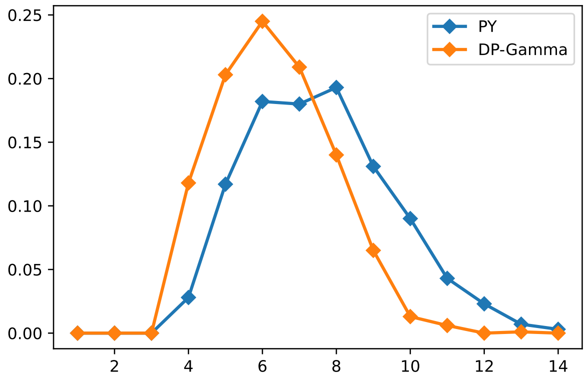


Figure 2.B.1: Posterior distribution for the number of clusters for the univariate dataset in Figure 2.B.1, under the PY prior and DPM prior with random concentration parameter.

1. Propose $\xi' \sim p(\xi'; \xi)$.
2. Generate an auxiliary variable $\boldsymbol{\mu}^{\text{aux}} \sim g(\boldsymbol{\mu} \mid \xi')/Z_{\xi'} \propto g(\boldsymbol{\mu} \mid \xi')$.
3. Accept ξ' with probability $\min\{1, \alpha^*\}$ where

$$\alpha^* \equiv \alpha^*(\xi; \xi' \mid \cdots) = \frac{p(\xi')g(\boldsymbol{\mu} \mid \xi')p(\xi; \xi')}{p(\xi)g(\boldsymbol{\mu} \mid \xi)p(\xi'; \xi)} \times \frac{g(\boldsymbol{\mu}^{\text{aux}} \mid \xi)}{g(\boldsymbol{\mu}^{\text{aux}} \mid \xi')}.$$

Comparing α^* to the acceptance ratio in (2.22), note that the ratio $Z_\xi/Z_{\xi'}$ has been replaced by a ratio of unnormalized densities, evaluated in the auxiliary variable $\boldsymbol{\mu}^{\text{aux}}$. The main difficulty is sampling $\boldsymbol{\mu}^{\text{aux}}$, which must follow the distribution of $\boldsymbol{\mu}$ given ξ'. To this end, we employ the stochastic dominated coupling from the past algorithm in Kendall and Møller (2000), which extends the coupling from the past algorithm in Propp and Wilson (1996) to uncountable partially ordered spaces. Specifically, we employed in our code Algorithm 11.7 in Møller and Waagepetersen (2004).

2.B Additional simulation studies

In addition to the simulation studies in Section 2.8, below we discuss different aspects of the M-w-G sampler and posterior inference.

2.B.1 Other competitors for Section 2.8.2

We consider here two further competitors for the simulation study in Section 2.8.2. Specifically, a Pitman-Yor process mixture with parameters $(1.0, 0.1)$ and a Dirichlet process mixture where the concentration parameter is random and, a priori, Gamma$(2, 2)$ distributed. Posterior inference for the number of clusters is reported in Figure 2.B.1.

2.B.2 Comparison of run-times and posterior inference when using DPP and Strauss process priors

For $q = 1, 2, \ldots, 5$, we simulated $n = 200$ observations from (2.24) with $\mu_0 = (-5, \ldots, -5)$, $\Sigma_0 = I_q$, $\omega = 1$, $\mu_1 = 5$, and $\sigma_1 = 1$. Then we applied our M-w-G sampler when the marginal prior for $\boldsymbol{\mu}$ is either the DPP or the Strauss process, with hyperparameters as in Section 2.7. Here, we considered two truncation levels for the approximation of the DPP

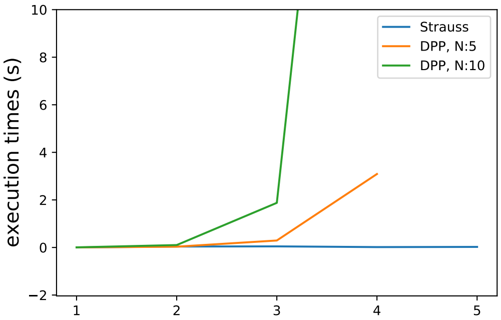


Figure 2.B.2: Per-iteration run-times as a function of data dimension q in case of DPP (with truncation levels $N = 5$ or 10) and Strauss process priors for $\boldsymbol{\mu}$.

density in (2.12), namely $N = 5$ and $N = 10$ (for comparison, Bianchini et al. (2020) suggested $N = 50$ when $q = 1$).

Figure 2.B.2 shows the per-iteration run-times of the M-w-G sampler as a function of the dimension q under either the DPP or Strauss process prior for $\boldsymbol{\mu}$. For each value of N, the computational cost associated to the DPP grows exponentially fast as the dimension q increases, unlike in the case of the Strauss process. In fact, the unnormalized density of the Strauss process is almost immediate to compute, and since the Strauss prior is quite informative on the number of components, cf. Section 2.7.1, the perfect simulation algorithm (see Section 2.6.3) does not impact significantly on the computational cost. Although not appreciable from Figure 2.B.2, the computational cost of our algorithm increases significantly with data dimension q also when we consider the Strauss process; in this case, the per-iteration computational cost goes from 0.0016 sec when $q = 1$ to 0.07 sec when $q = 5$, i.e., it increases by a factor of roughly 50.

As a further comparison, we simulated 500 univariate observations from model (2.23) and made again posterior computations under the Strauss process or the DPP prior for $\boldsymbol{\mu}$, where for the DPP density we fixed $\beta = 10$ (corresponding to the highest ESS in Table 2.1). For both cases of prior models, we ran the M-w-G sampler for $100,000$ iterations discarding the first $50,000$ as a burn-in and keeping one every ten iterations, for a final sample size of $5,000$. Figure 2.B.3 shows the true data generating density, together with Bayesian mixture density estimates and posterior distributions of the number of clusters under the two point process priors. Note that the two density estimates, as well as the two posterior distributions of the number of clusters, overlap almost perfectly. The Strauss process seems a good choice to model the prior of $\boldsymbol{\mu}$ since it, for a much smaller computational cost, provides same posterior summaries as the DPP.

2.B.3 Accuracy of cluster estimates

Figure 2.B.4 shows the posterior similarity matrices and the Adjusted Rand Index (ARI) scores for the univariate mixture of t and skew-normal distribution discussed in Section 2.8.2. The ARI is computed from the cluster labels $\boldsymbol{c}$ at each iteration of the MCMC chain as a measure of similarity between the estimated clusters and the true cluster. It is bounded by 1 and the larger value it assumes, the more similar is the estimated cluster to the true one. We report the posterior mean of the ARI ± one standard deviation on top of each posterior similarity matrix in Figure 2.B.4. The difference in the posterior similarity matrices is not so pronounced, but our repulsive mixture model gives the best ARI.

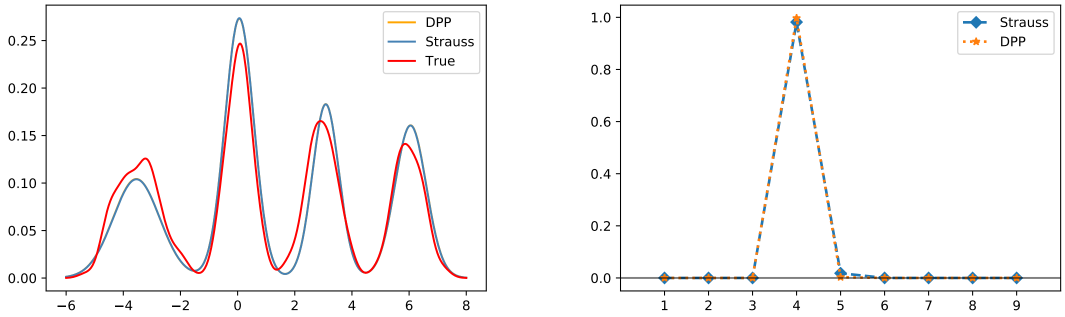


Figure 2.B.3: Bayesian mixture density estimates (left) and posterior distributions of the number of clusters (right) under the Strauss process (blue lines) and DPP (orange lines) priors for $\boldsymbol{\mu}$, together with the true mixture density which has four components. The orange lines overlap almost perfectly with the blue lines so that they are hardly visible.

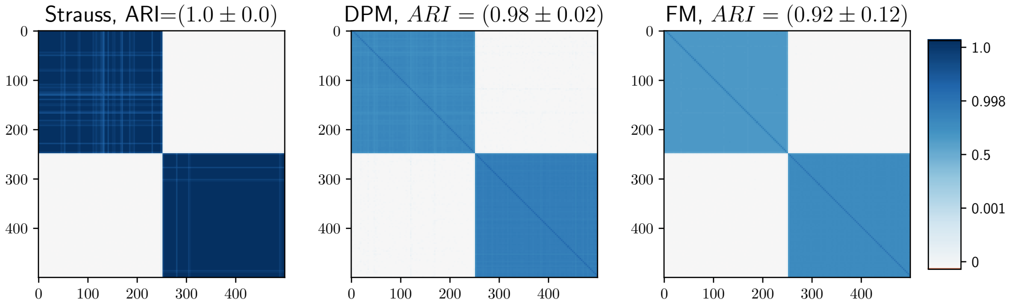


Figure 2.B.4: Posterior similarity matrices and ARI scores under the three models for the mixture of the univariate t and skew-normal distributions discussed in Section 2.8.2. The colors are on a logit scale to highlight differences around one.

2.B.4 The effect of the number of clusters

We consider how the number of clusters affects the performance of our M-w-G sampler. When $\boldsymbol{\mu}$ is distributed as the Strauss process, at every step of the MCMC algorithm a perfect simulation of $\boldsymbol{\mu}$ is required. The perfect simulation algorithm we use has a finite but random computational cost and, as argued in Section 2.6.3, it might become infeasible for a large number of clusters. On the other hand, when $\boldsymbol{\mu}$ is a DPP, the approximation of its density requires computing the determinant of the matrix C' in (2.10), which scales cubically with m. Furthermore, for the specific DPP considered in (2.12) computing C' requires the evaluation of $O(N^q m^2)$ inner products.

We generated $n = 500$ observations from a mixture of $m = 5, 9, 17, 25$ bivariate Gaussian densities, with locations given in Figure 2.B.5 (left), equal covariance matrices given by $0.5 I_2$, and with equal mixture weights. We compared the run-times (in seconds) required to complete 200 iterations with our M-w-G sampler when $\boldsymbol{\mu}$ is distributed either as the Strauss or the determinantal point process. Prior hyperparameters are fixed as in Section 2.7 (with $M_{\max} = 5m$) and Section 2.8. For the DPP, we considered two truncation levels of the spectral density, $N = 10, 50$. For each choice of m we generated 50 independent datasets and for the 200 M-w-G sampler iterations we used fixed and different independent random seeds.

In Figure 2.B.5 (right) for each m the run-times over the 50 independent datasets are denoted by dots, the median times by diamonds, and the median times are connected by

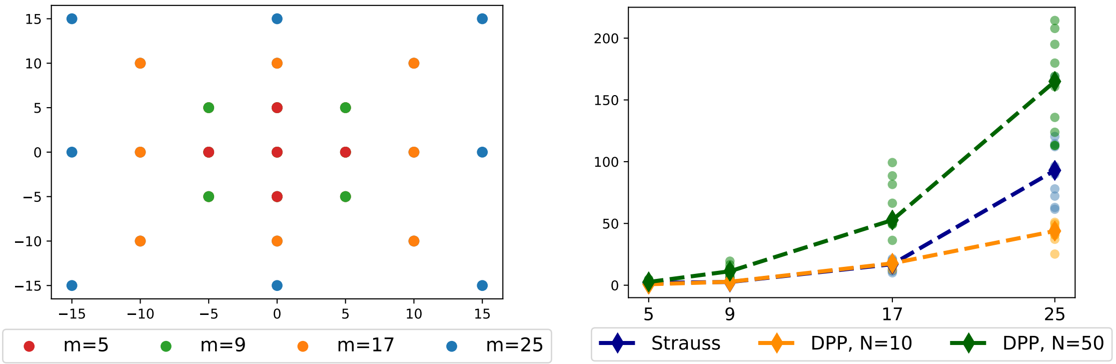


Figure 2.B.5: Locations of the true data generating process (left) and run-time comparison (right). The plot of the locations should be intended as follows: for $m = 5$ only the points labelled accordingly are considered, for $m = 9$ the points labeled as $m = 5$ and $m = 9$ are considered and so on. The run-times (in seconds) over 50 independently simulated datasets for each value of m are denoted by dots, we also report the median times as diamonds with a dashed line connecting them.

a dashed line. We see that the DPP with $N = 50$ is the most computationally demanding model for all values of m. When $m = 5, 9$, the Strauss process is significantly faster (up to 10 times faster) than the DPP with $N = 10$; instead, when $m = 17$, they have comparable computational costs. When $m = 25$, the perfect simulation algorithm starts to become more demanding; for example, the computational cost for the Strauss process is almost twice the one for the DPP with $N = 10$.

2.B.5 The effect of the data dimension

Below we compare our repulsive mixture model, the finite mixture model (FM) in Argiento and De Iorio (2022), and the Dirichlet process mixture model (DPM). See Section 2.8.2 for further details on how posterior inference is performed under the different models. In particular, we fix the hyper-parameters according to Sections 2.7 and 2.8.

f(Further, the curse of dimensionality, common to all clustering problems (Kriegel et al., 2009), implies a poor mixing of the algorithms. In addition to that, when considering a repulsive mixture model, things might be further complicated by either the need of perfect simulation to update possible hyperparameters ξ (when $\boldsymbol{\mu}$ follows the Strauss process) or the computation of the spectral density (when $\boldsymbol{\mu}$ follows the DPP given by (2.12)) which becomes prohibitive even for moderate values of q, as shown in Figure 2.B.2. f($q = 2, 5, 10, 15, 20, 25, 30$ increases. Moreover, we simulated $n = 200$ observations from

$$y_i \overset{\text{iid}}{\sim} 0.5\mathcal{N}(-5/\sqrt{q}\mathbf{1}_q, I_q) + 0.5\mathcal{N}(5/\sqrt{q}\mathbf{1}_q, I_q)$$

where $\mathbf{1}_q$ denotes the vector in $\mathbb{R}^q$ with elements all equal to one.

Table 2.B.1 reports posterior summaries as q increases for the three models. MCMC chains were run for $11,000$ iterations discarding the first $10,000$ as burn-in, so that the effective sample size must be referred to a total number of MCMC iterations equal to $1,000$. It is clear from Table 2.B.1 that as q increases, the mixing of the chains becomes progressively worse for all the models. In particular, the table shows the effective sample size (ESS) for the three cases: For our repulsive mixture model, the number of clusters k is constant for all the MCMC iterations when $q \geq 15$, and so ESS is zero; for FM, the ESS is zero when $q \geq 10$; and for DPM the ESS is zero for all values of q. The difference in the ARI scores is simply explained by the different strategy of initialization of the different

		$q = 5$	$q = 10$	$q = 15$	$q = 20$	$q = 25$	$q = 30$
Strauss	ARI	1.0	1.0	1.0	1.0	1.0	1.0
	ESS	240.3	250.1	0.0	0.0	0.0	0.0
	$\mathbb{E}[k \mid \text{data}]$	2.01	2.005	2.0	2.0	2.0	2.0
FM	ARI	1.0	1.0	1.0	1.0	1.0	1.0
	ESS	7.4	0.0	0.0	0.0	0.0	0.0
	$\mathbb{E}[k \mid \text{data}]$	2.01	2.0	2.0	2.0	2.0	2.0
DPM	ARI	1.0	0.0	0.0	0.0	0.0	0.0
	ESS	0.0	0.0	0.0	0.0	0.0	0.0
	$\mathbb{E}[k \mid \text{data}]$	2.00	1.0	1.0	1.0	1.0	1.0

Table 2.B.1: Adjusted Rand Index (ARI), effective sample size for the chain of the number of clusters k and posterior mean of k under the repulsive mixture model (Strauss), the non repulsive finite mixture model (FM) and the Dirichlet process mixture model (DPM).

software we ran: In our code for the M-w-G sampler, observations are initially randomly subdivided into 10 clusters; in the package `AntMAN`, which we used to fit the FM model, one cluster per observation is created; in the package `BNPMix`, used to fit the DPM model, all observations are initially allocated to one single cluster. In the latter case, the proposal of a new cluster is never accepted. Using our software or the package `AntMAN` instead, after a few MCMC iterations the observations are (correctly) partitioned into $k = 2$ clusters and no additional cluster is ever created.

Considering the effective sample size of k, Table 2.B.1 shows that repulsive mixture models might offer an advantage over non-repulsive mixture models when $q \leq 10$.

Perfect simulation is not a bottleneck here, as the number of points in the Strauss process is small. However, in one of several independent simulations, an unlucky initialization led to a large value of m in the first few iterations. As a consequence, the perfect simulation algorithm took longer to coalesce and indeed caused an out-of-memory problem on a 32 GB laptop.

2.C Removing the rectangular support assumption

Often we have assumed that the points of $\boldsymbol{\mu}$ have support given by a rectangular set R: For the theory in Sections 2.3–2.6, we made that assumption only for specificity and simplicity; in Section 2.8, we considered Gaussian mixture models and determined the rectangle R from the observations; while in Section 2.9, we considered the multivariate Bernoulli kernel and $R = [0, 1]^q$. Apart from the case of a DPP prior, it is often easy to modify everything without assuming R is rectangular and even compactness of R may be not be needed, In fact, the birth-death Metropolis-Hastings algorithm, which we always use to simulate the non-allocated process $\boldsymbol{\mu}^{(na)}$, can be specified in a very general setting, see Geyer and Møller (1994). On the other hand, for a DPP prior, compactness of R is needed when specifying a DPP density with respect to $\mathrm{d}\boldsymbol{\mu}$, and R needs to be a rectangle in order to use the spectral approach discussed in Lavancier et al. (2015). Recently, Poinas and Lavancier (2021) proposed a novel approximation of a general DPP density that does not require R to be rectangular (but still requires R is bounded).

3. Marked Point Processes for Bayesian Nonparametric Modelling

In this chapter, we study the distributional properties of the repulsive mixture model introduced in Chapter 2. To do so, we consider a simpler model, where we assume that we observe directly the latent process which generates the observation-specific parameters of the mixture. The study of this simpler model is carried out by means of Palm calculus and, in particular, we establish in this chapter several a priori properties of the model as well as a complete characterization of the marginal, posterior and predictive distributions. We then show that these quantities can be used either for prior elicitation, or as building blocks of two novel MCMC algorithms for posterior simulation. In this chapter, the treatment is more general than in Chapter 2. Specifically, we do not require that the process is absolutely continuous with respect to the unit-rate Poisson point process, which in turn allows us to extend our methodology to mixture models where the atoms exhibit a random clustering structure. We highlight the potential of this new class of mixtures via a simulation study.

3.1 Introduction

Random measures provide one of the main building blocks of Bayesian nonparametric inference, where they are used to directly model the observational process, e.g., in species sampling or feature sampling problems, or a latent unobserved process, e.g., in mixture models. They have also been adopted ass prior distribution for the hazard function in survival models.

In this chapter, we restrict our attention to random measures whose total mass is almost surely one. That is, we focus on random probability measures (RPMs). From the seminal work of Ferguson (Ferguson, 1973), where the celebrated Dirichlet process is introduced, a variety of approaches for the construction of RPMs have been introduced. A rather fruitful approach is based on the normalization of completely random measures with infinite activity, i.e., whose number of support points is countably infinite. To this end, recall that in Ferguson (1973) it is shown how the Dirichlet process can be constructed via the normalization of a Gamma random measure. This idea, systematically introduced in Regazzini et al. (2003) for measures on $\mathbb{R}$ with the name of normalized random measures with independent increments (NRMI), has been extended later to more general spaces. See, e.g., James et al. (2009) and the references therein. More recently, Argiento and De Iorio (2022) have exploited the same ideas to construct random probability measures with a random number of support points.

Starting from Petralia et al. (2012), several works have questioned the use of normalized completely random measures as prior distributions in mixture models, and, in particular, showing how assuming the atoms of the random measures to be independent (and typically identically distributed) leads to poor performance in model-based clustering. This behavior is not surprising. In fact, the consistency of mixture models for density estimation under very general data generating processes has been established, e.g., in Ghosal et al. (1999); Lijoi et al. (2005) for infinite mixture models and in Guha et al. (2021) for finite mixture models. Consider now the (very common) setting where one tries to fit a mixture of Gaussian densities to a dataset that has been generated from a mixture of k_0 non-Gaussian densities such as Student's t densities. Then, the posterior consistency for density estimation means that the mixture model will need $k \gg k_0$ Gaussian components to suitably approximate the true data generating density. Therefore, the posterior will find more than k clusters when the sample size is large enough.

In Beraha et al. (2022); Bianchini et al. (2020); Quinlan et al. (2020); Xie and Xu (2019); Xu et al. (2016), the authors assume a repulsive point process as joint prior for the support points of the random probability measures and their cardinality. In particular, in Beraha et al. (2022) it is empirically shown how such "repulsive mixtures" are more robust to misspecification compared to traditional non-repulsive mixtures.

3.1.1 Overview and Outline

Despite the recent popularity of repulsive mixtures, the statistical and probabilistic properties of random measures based on point processes other than the Poisson one have not been investigated. In this chapter, we propose a general construction for almost surely discrete random probability measures based on the normalization of *marked* point processes. Through the law of the point process, it is possible to encourage different behaviors among the support points of the random probability measure, such as independence (when the point process is Poisson or the class of IFPPs in Argiento and De Iorio (2022)), separation (i.e., the support points are well separated, when the point process is repulsive, such as Gibbs point processes or determinantal point processes), and also random aggregation (i.e., the support points are clustered together, when the point process is of Cox type). Our framework encompasses the priors previously proposed for repulsive mixture models, the models in Argiento and De Iorio (2022), and also the *mixture of mixtures* prior in Malsiner-Walli et al. (2017).

In this chapter, we establish distributional results for random probability measures built by normalizing marked point processes with general laws. These results are the counterpart of well-known results for NRMIs. Although our construction is general, we specialize our results to the case of Poisson, Gibbs, and Determinantal point processes throughout the discussion. The rest of the chapter is structured as follows. Section 3.2 introduces the statistical model and the general construction for normalized random measures based on marked point processes, and gives preliminaries on Palm calculus, a basic tool needed for all our proofs. In Section 3.3 we analyze the finite-dimensional distributions of our random measures. The posterior distribution of the random measure, the marginal distribution of the data, the distribution of the distinct values in the sample, and the predictive distribution of a new observation are discussed in Section 3.4. In Section 3.5 we show how to construct Bayesian mixture models based on our class of random probability measures, which act as the prior for the mixing measure of the mixture model. We discuss two computational algorithms to approximate the posterior distribution by simulation. Section 3.6 focuses on the class of shot-noise Cox process (Møller, 2003) and their use in mixture models. This class of point processes exhibits a random aggregation structure, and we suggest that this might be useful in the context of mixture models to perform clustering when the model is misspecified. The Appendix contains the proofs of our results and detailed calculations for all of the examples in the main text.

3.2 Model definition and preliminaries

Let us consider a sequence of random variables $(Y_i)_{i \geq 1}$ defined on the probability space $(\Omega, \mathcal{A}, \mathsf{P})$ and taking values in the Polish space $(\mathbb{X}, \mathcal{X})$, endowed with its Borel σ-algebra. We also indicate by $\boldsymbol{Y} := (Y_1, \ldots, Y_n)$ the observed sample of size n, with $n \geq 1$. Inferential conclusions are typically based on a kind of symmetry or analogy across data, for this reason we suppose that the sequence $(Y_i)_{i\geq 1}$ is exchangeable. Due to the de Finetti's representation theorem (de Finetti, 1937), exchangeability is equivalent to assuming that

$$\mathsf{P}\left(\bigcap_{i=1}^{n}\{Y_i \in B_i\}\right) = \int_{\mathbb{P}(\mathbb{X})} \prod_{i=1}^{n} p(B_i) Q(\mathrm{d}p) \tag{3.1}$$

for arbitrary Borel sets $B_1, \dots, B_n$ and $n \geq 1$. In (3.1), $\mathbb{P}(\mathbb{X})$ denotes the space of probability measures over $\mathbb{X}$, which is supposed to be endowed with its σ-algebra $\mathcal{P}(\mathbb{X})$. Finally, Q is a distribution over $(\mathbb{P}(\mathbb{X}), \mathcal{P}(\mathbb{X}))$, that is, the law of a random probability measure $\tilde{p}$. We may equivalently write (3.1) in a hierarchical fashion as follows:

$$\begin{aligned} Y_i \,|\, \tilde{p} &\overset{\text{iid}}{\sim} \tilde{p} \quad i \geq 1 \\ \tilde{p} &\sim Q. \end{aligned} \tag{3.2}$$

The de Finetti's theorem is important not only from a mathematical standpoint but also from a philosophical point of view, since it justifies the Bayesian approach to statistical inference. Indeed, Q in (3.1) plays the role of a prior distribution in the Bayesian setting. The choice of the prior Q has been the focus of several papers, starting from the seminal contribution of Ferguson (1973), who introduced the Dirichlet process (DP). Various generalizations of the DP have been proposed. A prominent class is the one of normalized random measures with independent increments (NRMIs, Regazzini et al., 2003), of which the DP is a special case. To recall the definition of NRMIs, consider the positive jumps of a subordinator $(S_j)_{j\geq 1}$ with Lévy intensity $\nu(s)\mathrm{d}s$, and mark each jump with a random variable X_i, defined on the space $(\mathbb{X}, \mathcal{X})$. The X_i's are here assumed to be i.i.d. from a diffuse distribution G_0. Hence, $\tilde{N} := \sum_{j\geq 1} \delta_{(S_j, X_j)}$ is Poisson random measure with intensity $\nu(s)\mathrm{d}sG_0(\mathrm{d}x)$ so that the measure

$$\tilde{\mu}(B) := \int_{\mathbb{R}_+ \times A} s\tilde{N}(\mathrm{d}s\,\mathrm{d}x), \qquad B \in \mathcal{X}$$

is completely random according to the definition of Kingman (1967). Then, $\tilde{p} := \tilde{\mu}/\tilde{\mu}(X)$ is a NRMI, provided that $\mathsf{P}(0 < \mu(\mathbb{X}) < +\infty) = 1$. See (Regazzini et al., 2003) for details.

Despite its generality, the class of NRMIs is limited to random measures based on the Poisson process. This may represent a limitation in some settings, as shown for instance in Cai et al. (2021) in the context of clustering in misspecified Bayesian mixture models.

In this chapter, we consider model (3.2) and propose a general construction for the random probability measure $\tilde{p}$, termed *normalized random measure* (nRM) that is based on general point processes other than the Poisson case. To construct a nRM, let Φ be a point process with points in $\mathbb{X}$ having distribution $\mathbf{P}_\Phi$. That is, $\Phi = \sum_{j\geq 1} \delta_{X_j}$ is a random counting measure over $(\mathbb{X}, \mathcal{X})$. Throughout the chapter, we will assume that Φ is *simple*, that is to say $X_i \neq X_j$ for $i \neq j$ almost surely. We refer to Daley and Vere-Jones (2003, 2008) for a thorough exposition of point processes and random measures. Then, we consider the marked point process Ψ on $\mathbb{X} \times \mathbb{R}_+$, $\Psi = \sum_j \delta_{(X_j, S_j)}$, constructed by assigning to the points in Φ independent and identically distributed (i.i.d.) marks with law H on the postive real line. Now define a random measure $\tilde{\mu}$ from Ψ as follows

$$\tilde{\mu}(B) = \int_{B \times \mathbb{R}_+} s\Psi(\mathrm{d}x\,\mathrm{d}s) = \sum_{j\geq 1} S_j \mathbb{1}_B(X_j), \qquad B \in \mathcal{X}. \tag{3.3}$$

Note that, when Φ is a Poisson point process, $\tilde{\mu}$ is completely random, i.e., for pairwise disjoint $B_1, \dots, B_n \in \mathcal{X}$, the random variables $\tilde{\mu}(B_1), \dots \tilde{\mu}(B_n)$ are mutually independent for any $n \geq 1$. In order to construct a random probability measure from $\tilde{\mu}$ in (3.3) we have to ensure $\mathsf{P}(0 < \tilde{\mu}(\mathbb{X}) < +\infty) = 1$. On the one hand, in order to have $\tilde{\mu}(\mathbb{X}) < +\infty$, it suffices to assume that Φ is a finite point process, which we will always do in the following, so that $\Phi(\mathbb{X}) < +\infty$ almost surely. See, e.g., Chapter 5 of Daley and Vere-Jones (2003) for an account of finite point processes. On the other hand, to ensure $\tilde{\mu}(\mathbb{X}) > 0$, we can condition on $\Phi(\mathbb{X}) \geq 1$. From a statistical point of view, we do not believe that it is essential to require $\tilde{\mu}(\mathbb{X}) > 0$. Indeed, the law of $\tilde{\mu}$ acts as a prior distribution in model (3.2) and we could set $\tilde{p} \equiv 0$ if $\tilde{\mu}(\mathbb{X}) = 0$, with the understanding that this improper likelihood means that the model does not generate any observations. As shown in Theorem 3.1 below, we have $\mathsf{P}(\tilde{\mu}(\mathbb{X}) = 0 \,|\, Y_1, \ldots, Y_n) = 0$ almost surely, so that the posterior distribution is always well behaved in this sense. Thus, $\tilde{\mu}$ can be employed to define a random probability measure by normalization $\tilde{p} := \tilde{\mu}/\tilde{\mu}(\mathbb{X})$, in particular we will write $\tilde{p} \sim \mathrm{nRM}(\mathbf{P}_\Phi; H)$ to denote the distribution of the normalized random measure $\tilde{\mu}$, and $\tilde{\mu} \sim \mathrm{RM}(\mathbf{P}_\Phi; H)$ for the distribution of the associated non-normalized random measure.

Here we discuss some important examples of point processes Φ, which will be widely used in the sequel to showcase the applicability of our results. In particular, besides Poisson processes, we also focus on Gibbs and Determinantal point processes.

Example 3.1 (Gibbs point processes)**.** *Following Baccelli et al. (2020), the probability distribution of a Gibbs-type point process has a density with respect to the law of a Poisson point process. More formally, let $\tilde{N}$ be a Poisson process on $(\mathbb{X}, \mathcal{X})$ with law $\mathbf{P}_N$, and let $f : \mathbb{X} \to \mathbb{R}_+$ be a measurable function with the additional requirement $\mathbb{E}[f(\tilde{N})] = 1$. Then we say that a point process Φ with distribution*

$$\mathbf{P}_\Phi(\mathrm{d}\nu) = f_\Phi(\nu)\mathbf{P}_N(\mathrm{d}\nu)$$

is a Gibbs point process with density f with respect to $\tilde{N}$. In the following, we will always consider densities with respect to the unit-rate Poisson point process on a compact subset R, that is $\tilde{N}$ has intensity measure $\nu(x) = \mathbb{1}_R(x)$.
In practice, constructing a density f is complex because of the constraint $\mathbb{E}[f_\Phi(\tilde{N})] = 1$. In particular, the expected value cannot be computed in closed form even for very simple densities. Therefore, usually one considers an unnormalized density g such that $\mathbb{E}[g(\tilde{N})] < +\infty$ and sets $f_\Phi(\nu) = g_\Phi/Z$ where $Z := \mathbb{E}[g_\Phi(\tilde{N})]$ is an intractable normalizing constant. See Møller and Waagepetersen (2004) for several examples of Gibbs point processes and associated unnormalized densities.

We will assume that Φ is hereditary, meaning that for any $\nu = \sum_{j=1}^m \delta_{X_j}$ and $X^* \in \mathbb{X} \setminus \{X_1, \ldots X_m\}$, $g(\nu + \delta_{X^*}) > 0$ implies $g(\nu) > 0$. Then we can introduce the so-called Papangelou conditional intensity defined as

$$\lambda_\Phi(\nu; \boldsymbol{X}^*) = \frac{f_\Phi\left(\nu + \sum_{j=1}^k \delta_{X_j^*}\right)}{f_\Phi\left(\sum_{j=1}^k \delta_{X_j^*}\right)} = \frac{g_\Phi\left(\nu + \sum_{j=1}^k \delta_{X_j^*}\right)}{g_\Phi\left(\sum_{j=1}^k \delta_{X_j^*}\right)} \tag{3.4}$$

where $\boldsymbol{X}^* = \{X_1^*, \ldots, X_k^*\}$. The Papangelou conditional intensity can be heuristically understood as the conditional "density" of Φ having atoms ν given that the rest of Φ is $\boldsymbol{X}^*$. If $\lambda_\Phi(\nu; \boldsymbol{X}^*)$ is a non-increasing function of $\boldsymbol{X}^*$, that is, $\lambda_\Phi(\nu; \boldsymbol{X}^*) \geq \lambda_\Phi(\nu; \boldsymbol{X}^* \cup X')$ for any $X' \in \mathbb{X} \setminus (\boldsymbol{X}^* \cup X')$, the point process Φ has a repulsive behavior.

The normalizing constant defining f_Φ is not relevant for Bayesian analyses if the (hyper)parameters appearing in the density g_Φ are fixed. If, instead, a hyperprior is assumed on them, updating those parameters is a so-called "doubly-intractable" problem. See, e.g., Beraha et al. (2022) for how to deal with hyperpriors in the context of repulsive Bayesian mixture models. We will consider hyperparameters fixed.

Example 3.2 (Determinantal point processes)**.** *Determinantal point processes (DPPs Macchi, 1975; Hough et al., 2006; Lavancier et al., 2015) are a class of repulsive point processes. We will restrict our focus to finite DPPs defined on a compact region $R \subset \mathbb{X} =: \mathbb{R}^q$. Consider a complex-valued covariance function $K : R \times R \to \mathbb{C}$ with spectral representation*

$$K(x, y) = \sum_{h \geq 1} \lambda_h \varphi_h(x) \overline{\varphi_h(y)}, \qquad x, y \in R \tag{3.5}$$

where $(\varphi_h)_{h\geq 1}$ for an orthonormal basis for $L^2(R)$, $\lambda_h \geq 0$ with $\sum_{h\geq 1} \lambda_h < +\infty$. The general construction of a DPP is given by introducing Bernoulli variables $B_h \sim Bern(\lambda_h)$ and considering $K'(x, y) = \sum_{h\geq 1} B_h \varphi_h(x)\overline{\varphi_h(y)}$. Then, conditional on $(B_h)_{h\geq 1}$, a DPP Φ with kernel K has density with respect to the unit-rate Poisson point process on R given by

$$f_{\Phi \,|\, (B_h)_h}(\nu) \propto \det \left\{ K'(x, y) \right\}_{x,y \in \nu}.$$

We remark that Φ consists of exactly $\sum_{h\geq 1} B_h$ points almost surely. The existence of a DPP with kernel K is equivalent to $\lambda_h \leq 1$ for any $h \geq 1$, see Macchi (1975).

If one specializes the construction above to the case $\lambda_h < 1$, it can be shown (cf. Lavancier et al. (2015)) that a DPP has density f_Φ with respect to the unit-rate Poisson process given by

$$f_\Phi(\nu) = e^{|R| - D} \det \left\{ C(x, y) \right\}_{x,y \in \nu}.$$

where $|\cdot|$ denotes the Lebesgue measure, $D := -\sum_{h\geq 1} \log(1 - \lambda_h)$, and

$$C(x, y) = \sum_{h \geq 1} \frac{\lambda_h}{1 - \lambda_h} \varphi_h(x) \overline{\varphi_h(y)}, \qquad x, y \in R.$$

We conclude the section with some elements of Palm calculus which is a fundamental tool to understand Bayesian analysis of nRM.

3.2.1 Palm distributions

Palm calculus is a basic tool in the study of point processes. For the analysis of random measures, the fundamental result needed from Palm theory is the Campbell-Little-Mecke formula, an extension of Fubini's theorem, which allows to exchange expectation and integral when considering expressions involving integrals of functionals of a point process Φ, where the integral is also with respect to the measure Φ. We provide background on Palm calculus in Appendix 3.A and refer the interested reader to Kallenberg (1984); Coeurjolly et al. (2017); Baccelli et al. (2020) for a detailed account. In the following, we limit ourselves to introduce the Palm measure of a point process and provide some intuition around it.

Let $\mathbb{M}(\mathbb{X})$ the space of boundedly finite measures on $\mathbb{X}$ and denote with $\mathcal{M}(\mathbb{X})$ its Borel σ-algebra. For a point process Φ on $\mathbb{X}$, let us introduce the mean measure M_Φ as $M_\Phi(B) := \mathbb{E}[\Phi(B)]$ for all $B \in \mathcal{X}$. We define the Campbell measure $\mathcal{C}_\phi$ on $\mathbb{X} \times \mathcal{M}(\mathbb{X})$ as

$$\mathcal{C}_\Phi(B \times L) = \mathbb{E}\left[\Phi(B) \mathbb{1}_L(\Phi)\right], \quad B \in \mathcal{X}, L \in \mathcal{M}(\mathbb{X}).$$

Then, as a consequence of the Radon-Nikodym theorem, there exists a M_Φ-a.e. unique disintegration probability kernel $\{\mathbf{P}_\Phi^x(\cdot)\}_{x\in\mathbb{X}}$ of $\mathcal{C}_\phi$ with respect to M_Φ, i.e.

$$\mathcal{C}_\Phi(B \times L) = \int_B \mathbf{P}_\Phi^x(L) M_\Phi(\mathrm{d}x) \quad B \in \mathcal{X}, L \in \mathcal{M}(\mathbb{X})$$

Note that, for any $x \in \mathbb{X}$, $\mathbf{P}_\Phi^x$ is the distribution of a random measure (specifically, a point process) on $\mathbb{X}$. Therefore, $\mathbf{P}_\Phi^x$ can be identified with the distribution of point process Φ_x

such that $\mathbf{P}^x_\Phi(L) = \mathsf{P}(\Phi_x \in L)$. In particular, $\mathbf{P}^x_\Phi$ can be understood as the law of the point process Φ, conditional to Φ having an atom at x. Following Baccelli et al. (2020) we call Φ_x the Palm version of Φ at x. Since δ_x is a trivial atom of Φ_x, we can subtract it from Φ_x and obtain the *reduced* Palm kernel, denoted by $\mathbf{P}^x_{\Phi^!}$, that is the law of the point process

$$\Phi^!_x := \Phi_x - \delta_x.$$

The argument outlined above can be extended to the case of multiple points $\boldsymbol{x} = (x_1, \dots, x_k)$, leading to the k-th Palm distribution $\{\mathbf{P}^{\boldsymbol{x}}_\Phi\}_{\boldsymbol{x}\in\mathbb{X}^n}$. Again, $\Phi_{\boldsymbol{x}}$ can be understood as the law of Φ conditional to Φ having atoms at $\{x_1, \dots, x_k\}$ and removing the trivial atoms yields the reduced Palm distribution that is the law of

$$\Phi^!_{\boldsymbol{x}} := \Phi_{\boldsymbol{x}} - \sum_{j=1}^{k} \delta_{x_j}.$$

It is easy to show that, for a marked point process Ψ, with independent marks, the Palm distribution $\Psi_{\boldsymbol{x},\boldsymbol{s}}$, with $\boldsymbol{x} = (x_1, \dots, x_k)$ and $\boldsymbol{s} = (s_1, \dots, s_k)$, does not depend on $\boldsymbol{s}$. Moreover, $\Psi^!_{\boldsymbol{x},\boldsymbol{s}}$ has the same law of the point process obtained by considering $\Phi^!_{\boldsymbol{x}}$ and marking it with i.i.d. marks. See Lemma 3.2 in the Appendix. Hence, we can write $\Psi^!_{\boldsymbol{x}}$ in place of $\Psi^!_{\boldsymbol{x},\boldsymbol{s}}$ and define

$$\widetilde{\mu}^!_{\boldsymbol{x}}(A) := \int_{A\times\mathbb{R}_+} s\Psi^!_{\boldsymbol{x}}(\mathrm{d}x\,\mathrm{d}s) \sim \mathbf{P}^{\boldsymbol{x}}_{\Psi^!} \tag{3.6}$$

See Appendix 3.A for further details on Palm calculus. For our analyses, Palm calculus is essential because, thanks to the Campbell-Little-Mecke formula, it allows us to manipulate the expected values of integrals with respect to $\widetilde{\mu}$, where both the integral and the expected value are with respect to $\widetilde{\mu}$.

3.3 Prior analysis

In this section and in the remainder of the chapter, we focus on the model (3.2), by choosing a general nRM, that is, $\tilde{p} \sim \mathrm{nRM}(\mathbf{P}_\Phi; H)$ and $\tilde{p}$ is obtained by the normalization of a random measure $\tilde{\mu}$. Before moving to the posterior analysis, it is often useful to investigate finite dimensional statistics induced by the law of $\widetilde{\mu}$ for prior elicitation. Here, we provide some results along these lines, namely, characterizing expectations $\mathbb{E}[\widetilde{\mu}(A)]$ and $\mathbb{E}[\tilde{p}(A)]$, the covariance $\mathrm{Cov}(\widetilde{\mu}(A), \widetilde{\mu}(B))$ and the correlation between $\tilde{p}(A)$ and $\tilde{p}(B)$.

Proposition 3.1. *Let $\tilde{\mu} \sim \mathrm{RM}(\mathbf{P}_\Phi; H)$. Let us denote by $\mathbb{E}[S] := \int_{\mathbb{R}_+} sH(\mathrm{d}s)$ the expected value of a random variable $S \sim H$. Then, we have:*

(i) for any measurable set A, $\mathbb{E}[\widetilde{\mu}(A)] = M_\Phi(A)\mathbb{E}[S]$, and

$$\mathbb{E}[\tilde{p}(A)] = \int_{\mathbb{R}_+} \psi(u) \int_A \mathbb{E}\left[e^{-u\widetilde{\mu}^!_x(\mathbb{X})}\right] M_\Phi(\mathrm{d}x)\mathrm{d}u$$

where $\psi(u) := \mathbb{E}[e^{-uS}]$;

(ii) the covariance equals

$$Cov(\widetilde{\mu}(A), \widetilde{\mu}(B)) = (M_{\Phi^2}(A\times B) - M_\Phi(A)M_\Phi(B))\,\mathbb{E}[S]^2,$$

for arbitrary measurable sets $A, B \in \mathcal{X}$.

We now specialize the previous proposition in some examples of interest.

Example 3.3 (Poisson process)**.** *If Φ is a Poisson point process with intensity $\nu(\mathrm{d}x)$, $M_\Phi(A) = \nu(A)$ and $M_{\Phi^2}(A \times B) = \nu(A)\nu(B) + \nu(A \cap B)$. Hence*

$$\mathbb{E}[\widetilde{\mu}(A)] = \mathbb{E}[S]\nu(A), \qquad Cov(\widetilde{\mu}(A), \widetilde{\mu}(B)) = \mathbb{E}[S]^2\nu(A \cap B)$$

In particular, if $A \cap B = \emptyset$, $Cov(\widetilde{\mu}(A), \widetilde{\mu}(B))$ is zero (in fact, the random variables are independent).

Example 3.4 (Gibbs point process)**.** *If Φ is a Gibbs point process with density f_Φ:*

$$M_{\Phi^k}(\mathrm{d}x_1 \cdots \mathrm{d}x_k) = \ E_{\tilde{N}}\left[f_\Phi\left(\tilde{N} + \sum_{j=1}^{k} \delta_{x_j}\right)\right].$$

In this case, the moment measure cannot be usually be computed in closed form. However, the expected value can be efficiently approximated numerically via Monte Carlo simulations.

Example 3.5 (Determinantal point process)**.** *If Φ is a DPP, $M_\Phi(A) = \int_A K(x,x)\mathrm{d}x$. Moreover, exploiting the relation between the moment and factorial moment measures,*

$$M_{\Phi^2}(A \times B) = \int_{A \times B} \det\{K(x_i, x_j)\}_{i,j=1}^2 \mathrm{d}x_1\mathrm{d}x_2 + M_\Phi(A \cap B)$$

When $K(x,y) = \rho \exp^{-\|x-y\|^2/\alpha}$ we recover the so-called Gaussian-DPP. In this case $M_\Phi(A) = \rho|A|$, where $|\cdot|$ is the Lebesgue measure of a set, and

$$M_{\Phi^2}(A \times B) = \rho^2 \int_{A \times B} 1 - e^{-2\|x-y\|^2/\alpha}\mathrm{d}x\mathrm{d}y + \rho|A \cap B|.$$

3.4 Bayesian analysis of nRMs

This section contains the most relevant results of the chapter: posterior, marginal and predictive distributions for the statistical model (3.2), when $\tilde{p} \sim \mathrm{nRM}(\mathbf{P}_\Phi; H)$. All the results are available in closed form and they constitute the backbone to develop computational procedures in presence of the new class of priors.

All the proofs of these important theoretical achievements are based on the Laplace functional of the random measure $\widetilde{\mu}$, defined as

$$L_{\widetilde{\mu}}(f) = \mathbb{E}\left[\exp\left(-\int_{\mathbb{X}} f(x)\widetilde{\mu}(\mathrm{d}x)\right)\right]$$

for any bounded non-negative function $f : \mathbb{X} \to \mathbb{R}_+$ of bounded support. We now introduce some useful notations and an additional variable U_n which helps us to describe the Bayesian analysis of the model in (3.2). Denote by $\mathbf{P}_\Psi$ the distribution induced on $\widetilde{\mu}$ by the law of the marked point process Ψ, the joint distribution of $(\boldsymbol{Y}, \widetilde{\mu})$ is

$$\mathsf{P}(\boldsymbol{Y} \in \mathrm{d}\boldsymbol{y}, \tilde{\mu} \in \mathrm{d}\mu) = \frac{1}{T^n}\prod_{i=1}^{n} \mu(\mathrm{d}y_j)\mathbf{P}_\Psi(\mathrm{d}\mu) \tag{3.7}$$

where $T = \mu(\mathbb{X})$. As in James et al. (2009), we introduce an auxiliary variable $U_n \,|\, T \sim \mathrm{Gamma}(n, T)$ and, by a suitable augmentation of the underlying probability space, we now consider the joint distribution of $(\boldsymbol{Y}, U_n, \tilde{\mu})$

$$\mathsf{P}(\boldsymbol{Y} \in \mathrm{d}\boldsymbol{y}, U_n \in \mathrm{d}u, \tilde{\mu} \in \mathrm{d}\mu) = \frac{u^{n-1}}{\Gamma(n)}e^{-Tu}\mathrm{d}u\prod_{i=1}^{n} \mu(\mathrm{d}y_j)\mathbf{P}_\Psi(\mathrm{d}\mu).$$

Since $\widetilde{\mu}$ is almost surely discrete, with positive probability there will be ties within the sample $\boldsymbol{Y} = (Y_1, \dots, Y_n)$. For this reason $\boldsymbol{Y}$ is equivalently characterized by the couple $(\boldsymbol{Y}^*, \tilde{\pi})$, where $\boldsymbol{Y}^* = (Y_1^*, \dots, Y_k^*)$ is the vector of distinct values and $\tilde{\pi}$ is a random partition of $[n] := \{1, \dots, n\}$ of size K_n, which contains the observations within the sample that are equal. Given $K_n = k$, we indicate by $\boldsymbol{n} = (n_1, \dots, n_k)$ the vector of counts, i.e., n_j is the cardinality of the set $\{i \in [n] : Y_i = Y_j^*\}$, , as $j = 1, \dots, k$. As a consequence we may write:

$$\mathsf{P}(\boldsymbol{Y} \in \mathrm{d}\boldsymbol{y}, U_n \in \mathrm{d}u, \tilde{\mu} \in \mathrm{d}\mu) = \frac{u^{n-1}}{\Gamma(n)} e^{-Tu} \prod_{j=1}^{k} \mu(\mathrm{d}y_j^*)^{n_j} \mathbf{P}_\Psi(\mathrm{d}\mu).$$

3.4.1 Posterior characterization

We first characterize the posterior distribution of $\tilde{p} \sim \mathrm{nRM}(\mathbf{P}_\Phi; H)$, when this is employed in (3.2). Since $\tilde{p}$ is obtained by the normalization of the random measure $\tilde{\mu}$, it is sufficient to describe the posterior distribution of $\tilde{\mu}$, which is provided by the following.

Theorem 3.1. *Assume that $(Y_i)_{i \geq 1}$ is an exchangeable sequence of observations as in (3.2), where $\tilde{p} \sim \mathrm{nRM}(\mathbf{P}_\Phi; H)$ and it arises as the normalization of the random measure $\widetilde{\mu}$. Assume that $H(\mathrm{d}s) = h(s)\mathrm{d}s$ where $\mathrm{d}s$ is the Lebesgue measure. The distribution of $\widetilde{\mu}$ conditionally on $\boldsymbol{Y} = \boldsymbol{y}$ and U_n coincides with the one of the random measure*

$$\sum_{j=1}^{k} S_j^* \delta_{Y_j^*} + \widetilde{\mu}' \tag{3.8}$$

where:

(i) $\boldsymbol{S}^* := (S_1^*, \dots, S_k^*)$ is a vector of independent random variables, and the density of S_j^* equals

$$f_{S_j^*}(s) \propto e^{-U_n s} s^{n_j} h(s), \ \ \textit{as } j = 1, \dots, k;$$

(ii) $\widetilde{\mu}'$ is a random measure with Laplace functional

$$\mathbb{E}\left[\exp \int_{\mathbb{X}} -f(z)\widetilde{\mu}'(\mathrm{d}z)\right] = \frac{\mathbb{E}\left[\exp\left\{-\int_{\mathbb{X}}(f(z) + U_n)\widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)\right\}\right]}{\mathbb{E}\left[\exp\left\{-\int_{\mathbb{X}} U_n \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)\right\}\right]}, \tag{3.9}$$

where $\widetilde{\mu}^!_{\boldsymbol{y}^*}$ is as in (3.6) for $\boldsymbol{x} = \boldsymbol{y}^*$.

Finally, the conditional distribution of U_n given $\boldsymbol{Y} = \boldsymbol{y}$ has a density with respect to the Lebesgue measure proportional to

$$f_{U_n \mid \boldsymbol{Y}}(u) \propto u^{n-1} \mathbb{E}\left[e^{-\int u \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right] \prod_{j=1}^{k} \kappa(u, n_j) \mathbb{1}_{\mathbb{R}_+}(u) \tag{3.10}$$

with $\kappa(u, n) := \int_{\mathbb{R}_+} e^{-us} s^n H(\mathrm{d}s)$.

The expression in (3.9) is obtained without any specific assumption on the law of the point process Φ. More intuitive expressions for the posterior distribution of $\widetilde{\mu}$ are obtained specializing the expression to particular classes of point processes. In the following, we denote by $\psi(u) := \mathbb{E}[e^{-uS}]$ the Laplace transform of $S \sim H$ evaluated at $u \in \mathbb{R}_+$. Moreover, we define

$$f^u_{S'}(s) := \frac{e^{-su} h(s)}{\int_{\mathbb{R}^+} e^{-su} h(s)(\mathrm{d}s)}. \tag{3.11}$$

to be the density of the exponential tilting of S.

Example 3.6 (Poisson point process). *If Φ is a Poisson point process with intensity $\nu(x)\mathrm{d}x$, the random measure $\widetilde{\mu}'$ in Theorem 3.1 equals the distribution of*

$$\sum_{j\geq 1} S_j'\delta_{X_j'}$$

where $\Psi' := \sum_{j\geq 1}\delta_{(X_j',S_j')}$ is a marked point process whose unmarked point process $\Phi' := \sum_{j\geq 1}\delta_{X_j'}$ is a Poisson process with intensity given by $\psi(u)\nu(\mathrm{d}x)$ and the marks S_j' are i.i.d. with distribution (3.11). Hence $\widetilde{\mu}'$ is a completely random measure. Moreover, by Lemma 3.3 in the Appendix,

$$\mathbb{E}[e^{-\int u\widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}] = \mathbb{E}[\exp(\log\psi(u)\Phi^!(\mathbb{X}))] = e^{\nu(\mathbb{X})(\psi(u)-1)}$$

where the last equality follows from the fact that $\Phi^!(\mathbb{X})$ is a Poisson random variable with parameter $\nu(\mathbb{X})$. We conclude by noting that, in this case, $\widetilde{\mu}'$, given U_n, does not depend on the observed sample.

Example 3.6 can be generalized to a broader class of point processes Φ where the atoms, conditionally to the number of points in the process, are independent random variables. Random measures obtained by marking and normalizing this kind of processes have been studied in Argiento and De Iorio (2022) under the name of normalized independent finite point processes (Norm-IFPP). We can consider this class of measures as a special case of the ones under investigation here. Now we discuss other examples, whose posterior representation is still not available in the Bayesian nonparametric literature.

Example 3.7 (Gibbs point process). *If Φ is a Gibbs point process with density f_Φ, the random measure $\tilde{\mu}'$ equals the distribution of*

$$\sum_{j\geq 1} S_j'\delta_{X_j'}$$

where $\Psi' := \sum_{j\geq 1}\delta_{(X_j',S_j')}$ is a marked point process whose unmarked point process $\Phi' := \sum_{j\geq 1}\delta_{X_j'}$ is of Gibbs type with density with respect to the Poisson process $\tilde{N}$ given by

$$f_{\Phi'}(\nu) := \frac{\exp\left\{\int_{\mathbb{X}}\log\left(\int_{\mathbb{R}_+} e^{-sU_n}H(\mathrm{d}s)\right)\nu(\mathrm{d}x)\right\} f_\Phi(\nu + \sum_{j=1}^k \delta_{Y_j^*})}{\mathbb{E}_N\left[\exp\left\{\int_{\mathbb{X}}\log\left(\int_{\mathbb{R}_+} e^{-sU_n}H(\mathrm{d}s)\right)N(\mathrm{d}x)\right\} f_\Phi(N + \sum_{j=1}^k \delta_{Y_j^*})\right]},$$

and the marks $\tilde{S}_j'$ are i.i.d. with distribution (3.11). The denominator in the last expression seems rather daunting to compute. This is usually the case for Gibbs point process, so it should come as no surprise. Considering the unnormalized density

$$f_{\Phi'}(\nu) \propto q_{\Phi'}(\nu) = \exp\left\{\int_{\mathbb{X}}\log\left(\int_{\mathbb{R}_+} e^{-sU_n}H(\mathrm{d}s)\right)\nu(\mathrm{d}x)\right\} f_\Phi(\nu + \sum_{j=1}^k \delta_{Y_j^*})$$

and denoting with n_ν the cardinality of ν, the exponential can be written as

$$\exp\left\{\int_{\mathbb{X}}\log\left(\int_{\mathbb{R}_+} e^{-sU_n}H(\mathrm{d}s)\right)\nu(\mathrm{d}x)\right\} = \psi^{n_\nu}(U_n),$$

Finally, recalling the definition of Papangelou conditional intensity in (3.4), the unnormalized density $q_{\Phi'}$ can be then expressed as

$$q_{\Phi'}(\nu) = \psi^{n_\nu}(U_n)\lambda_\Phi(\nu;\boldsymbol{Y}^*)$$

which does not depend on the intractable normalizing constant in f_Φ, see (3.4)*. This is particular useful for posterior simulation (cf. Section 3.5.1) since several algorithms for simulation from unnormalized point process densities are available. Although it is not explicitly stated, an equivalent result to the one above (albeit obtained in a different way) is at the core of the MCMC sampling scheme proposed in Beraha et al. (2022).*

Example 3.8 (Determinantal point process)**.** *Assume that Φ is a DPP with kernel K. Moreover assume that its eigenvalues λ_j in* (3.5) *are all strictly smaller than one. Then, the random measure $\widetilde{\mu}'$ equals the distribution of*

$$\sum_{j\geq 1} S'_j \delta_{X'_j}$$

where $\Psi' := \sum_{j\geq 1} \delta_{(X'_j, S'_j)}$ is a marked point process whose unmarked point process $\Phi' := \sum_{j\geq 1} \delta_{X'_j}$ is a DPP with density with respect to the unit rate Poisson process on $R \subset \mathbb{X} =: \mathbb{R}^q$ given by $f_{\Phi'}(\nu) \propto \det[C'(x_i, x_j)]_{(x_i,x_j)\in\nu}$, where, conditionally on $\boldsymbol{Y}^* = \boldsymbol{y}^*$, we have

$$C'(x,y) = \psi(u)\left[C(x,y) - \sum_{i,j=1}^{k} \left(C_{\mathbf{y}^*}^{-1}\right)_{i,j} C(x, y_i^*)C(y, y_j^*)\right],$$

and the marks $\tilde{S}'_j$ are i.i.d. with distribution (3.11)*. Note that for simulation purposes, it is useful to know or approximate the eigendecomposition of the kernel K' associated to the DPP Φ' (i.e., the kernel C' is to K' as the kernel C is to K in Section 3.2). K' of Φ' can be deduced from the eigendecomposition of C'. Writing*

$$C'(x,y) = \sum_j \gamma_j \varphi'_j(x)\overline{\varphi'}_j(y),$$

we have that

$$K'(x,y) = \sum_j \lambda'_j \varphi'_j(x)\overline{\varphi'}_j(y),$$

where $\lambda'_j = \gamma_j/(1+\gamma_j)$ and the γ_j's and φ'_j's can be approximated numerically using the Nyström method (Sun et al., 2015)

3.4.2 Marginal and predictive distributions

In the present section we describe the marginal and predictive distributions. We start with the first one.

Theorem 3.2. *Assume that $(Y_i)_{i\geq 1}$ is an exchangeable sequence of observations as in* (3.2)*, where $\tilde{p} \sim \mathrm{nRM}(\mathbf{P}_\Phi; H)$. The marginal distribution of a sample $\boldsymbol{Y}$ exhibiting $K_n = k$ distinct values $\boldsymbol{Y}^*$ with respective counts $n_1, \ldots, n_k$ equals*

$$\mathsf{P}(\boldsymbol{Y} \in \mathrm{d}\boldsymbol{y}) = \int_{\mathbb{R}_+} \frac{u^{n-1}}{\Gamma(n)} \mathbb{E}\left[e^{-\int_{\mathbb{X}} u\widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right] \prod_{j=1}^{k} \kappa(u, n_j)\mathrm{d}u\, M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*),$$

where $\widetilde{\mu}^!_{\boldsymbol{y}^*}$ is defined in (3.6) *and M_{Φ^k} is the k-th moment measure of Φ.*

We now specialize Theorem 3.2 in some important examples.

Example 3.9 (Poisson point process, cont'd)**.** *First, note that, since the values $y_1^*, \ldots, y_k^*$ are pairwise distinct, one has $M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*) = \prod_{i=1}^k \nu(\mathrm{d}y_i^*)$. Hence, conditionally to $U_n = u$, the marginal law in Theorem 3.2 is proportional to the following quantity*

$$e^{\nu(\mathbb{X})(\psi(u)-1)} \prod_{j=1}^{k} \kappa(u, n_j) \prod_{j=1}^{k} \nu(\mathrm{d}y_i^*).$$

Example 3.10 (Gibbs point process, cont'd)**.** *Let Φ the Gibbs point process defined in Example 3.7, denote with λ the intensity measure of $\tilde{N}$ (the Poisson point process with respect to which the density is specified)*

$$M_{\Phi^k}(B) = \; E_{\tilde{N}}\left[f_\Phi\left(\tilde{N} + \sum_{j=1}^{k} \delta_{x_j}\right)\right] \lambda^k(B), \qquad B \in \mathcal{X}^k \tag{3.12}$$

and, in particular, conditionally to $U_n = u$ the marginal distribution of data is proportional to

$$\prod_{j=1}^{k} \kappa(u, n_j) \mathbb{E}\left[\exp\left(\int_{\mathbb{X}} \log \psi(u) \tilde{N}(\mathrm{d}x)\right) f_\Phi(\tilde{N} + \sum_{j=1}^{k} \delta_{y_j^*})\right].$$

Example 3.11 (Determinantal point process, cont'd)**.** *If Φ is a DPP, for all $x_1, \ldots, x_n$ such that $K(x_j, x_j) > 0$, $\Phi^!_{\boldsymbol{x}}$ is a DPP with kernel as in Example 3.8. Then, $\mathbb{E}[e^{-\int_{\mathbb{X}} u\widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}] = \psi(u)^{\sum_{j\geq 1} \lambda'_j}$. Moreover, since $y_1^*, \ldots, y_k^*$ are pairwise different, we have that the moment measure M_{Φ^k} equals to the factorial moment measure $M_{\Phi^{(k)}}$, which in the case of a DPP has explicit expression*

$$M_{\Phi^{(k)}}(\mathrm{d}\boldsymbol{y}^*) = \det\{K(y_i^*, y_j^*)\}_{i,j=1}^k \mathrm{d}\boldsymbol{y}^*$$

where $\mathrm{d}\boldsymbol{y}^*$ denotes the k-fold Lebesgue measure.

We now focus on the derivation of the predictive distribution, i.e., the distribuion of Y_{n+1}, conditionally on the observable sample $\boldsymbol{Y}$. Let us first define the probability of a latent variable U_n, given a data point $\boldsymbol{Y} = \boldsymbol{y}$,

$$f_{U_n}(u|\boldsymbol{y}) := \frac{u^{n-1}\mathbb{E}\left[e^{-\int u\widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right] \prod_{j=1}^k \kappa(u, n_j)}{\int_{\mathbb{R}_+} u^{n-1}\mathbb{E}\left[e^{-\int u\widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right] \prod_{j=1}^k \kappa(u, n_j)\mathrm{d}u} \tag{3.13}$$

which is a density on $\mathbb{R}_+$. We further assume that there exists a non-atomic probability probability measure on $(\mathbb{X}, \mathcal{X})$ such that the measure $M_{\Phi_k}(\,\cdot\,)$ is absolutely continuous with respect to the product measure $P_0^k(\,\cdot\,)$. The Radon-Nikodym derivative of M_{Φ_k} is denoted by m_{Φ_k}.

Theorem 3.3. *Assume that $(Y_i)_{i\geq 1}$ is an exchangeable sequence of observations as in* (3.2)*, where $\tilde{p} \sim \mathrm{nRM}(\mathbf{P}_\Phi; H)$. Moreover, suppose that $M_{\Phi_k} \ll P_0^k$, for some probability measure P_0, with Radon-Nikodym derivative m_{Φ_k}. Then, the distribution of Y_{n+1}, conditionally on $\boldsymbol{Y} = \boldsymbol{y}$ and U_n with density* (3.13)*, equals*

$$\begin{aligned}\mathsf{P}(Y_{n+1} \in A \,|\, \boldsymbol{Y} = \boldsymbol{y}, U_n) \propto \sum_{j=1}^{k} \frac{\kappa(U_n, n_j + 1)}{\kappa(U_n, n_j)} \delta_{Y_j^*}(A) \\ + \int_A \kappa(U_n, 1) \frac{\mathbb{E}\left[e^{-\int_{\mathbb{X}} U_n \widetilde{\mu}^!_{(\boldsymbol{y}^*, y)}(\mathrm{d}z)}\right]}{\mathbb{E}\left[e^{-\int_{\mathbb{X}} U_n \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right]} \frac{m_{\Phi^{k+1}}(\boldsymbol{y}^*, y)}{m_{\Phi^k}(\boldsymbol{y}^*)} P_0(\mathrm{d}y),\end{aligned}$$

where $A \in \mathcal{X}$.

The predictive distribution from Theorem 3.3 can be interpreted using a generalized Chinese restaurant process metaphor. The first customer arrives at the restaurant and sits at the first table, eating dish $Y_1 = Y_1^*$ such that $\mathsf{P}(Y_1^* \in \mathrm{d}y) \propto M_\Phi(\mathrm{d}y)$. The second customer sits at the same table of the first customer with probability proportional to $\kappa(U_1, 2)/\kappa(U_1, 1)$, or sits at a new table and eats a new dish with probability proportional to

$$\begin{aligned}&\mathsf{P}(Y_2 \in \mathbb{X} \setminus \{y_1^*\} \,|\, Y_1^* = y_1^*)\\ &\quad\propto \frac{\kappa(U_1, 1)}{\mathbb{E}\left[e^{-\int_{\mathbb{X}} U_1 \widetilde{\mu}^!_{y_1^*}(\mathrm{d}z)}\right] m_\Phi(y_1^*)} \int_{\mathbb{X}} \mathbb{E}\left[e^{-\int_{\mathbb{X}} U_1 \widetilde{\mu}^!_{(y_1^*, y)}(\mathrm{d}z)}\right] m_{\Phi^2}(y_1^*, y) P_0(\mathrm{d}y),\end{aligned}$$

where $U_1 \sim f_{U_1 \,|\, y_1}$ defined in (3.10). The distribution of the new dish is proportional to

$$\mathsf{P}(Y_2 \in \mathrm{d}y | Y_1^* = y_1^*) \propto \mathbb{E}\left[e^{-\int_{\mathbb{X}} u \widetilde{\mu}^!_{(y_1^*, y)}(\mathrm{d}z)}\right] m_{\Phi^2}(y_1^*, y) P_0(\mathrm{d}y).$$

The metaphor proceeds as usual for new customers entering the restaurant at time n. As in the traditional Chinese Restaurant process, customers sitting at the same table eat the same dish, whereas the same dish cannot be served at different tables. We now specialize the predictive distribution in some examples of interest.

Example 3.12 (Poisson point process, cont'd)**.** *It is clear that*

$$\frac{m_{\Phi^{k+1}}(\boldsymbol{y}^*, y)}{m_{\Phi^k}(\boldsymbol{y}^*)} P_0(\mathrm{d}y) = \nu(\mathrm{d}y)$$

since in the Poisson process case the k-th moment measure, when evaluated at disjoint sets, factorizes. Moreover, by the properties of the Poisson point process, we have that $\mu^!_{(\boldsymbol{y}^*, y)}$ and $\mu^!_{(\boldsymbol{y}^*)}$ have the same distribution, so that the ratio of expectations in Theorem 3.3 disappears. We remark that, in general, this is no longer the case for the more general Norm-IFFP processes in Argiento and De Iorio (2022) and more care has to be taken in deriving the distribution of the number of points in $\mu^!_{(\boldsymbol{y}^*, y)}$ and $\mu^!_{\boldsymbol{y}^*}$.

Example 3.13 (Gibbs point process, cont'd)**.** *Using notation from Example 3.10, it is clear that $M_{\Phi^k} \ll P_0^k$, where P_0 is obtained by normalizing some measure λ. In most applications, λ is taken to be the Lebesgue measure on a compact set, as a consequence P_0 is the probability density function of a uniform random variable. The predictive distribution in Theorem 3.3 boils down to*

$$\begin{aligned}\mathsf{P}(Y_{n+1} \in A \,|\, \boldsymbol{Y} = \boldsymbol{y}, U_n) \propto &\sum_{j=1}^{k} \frac{\kappa(U_n, n_j + 1)}{\kappa(U_n, n_j)} \delta_{y_j^*}(A)\\ &+ \int_A \kappa(U_n, 1) \frac{\mathbb{E}\left[\psi^{\tilde{N}(\mathbb{X})}(U_n) \lambda_\Phi(\tilde{N}; \sum_{j=1}^k \delta_{y_j^*} + \delta_y)\right]}{\mathbb{E}\left[\psi^{\tilde{N}(\mathbb{X})}(U_n) \lambda_\Phi(\tilde{N}; \sum_{j=1}^k \delta_{y_j^*})\right]} \lambda(\mathrm{d}y).\end{aligned} \tag{3.14}$$

where the last term involves two expected values with respect to the Poisson process $\tilde{N}$, which can be easily approximated via Monte Carlo integration, since it is generally straightforward to sample from the law of $\tilde{N}$.

Example 3.14 (Determinantal point process, cont'd)**.** *Let C', K' be defined as in Example 3.8, with associated eigenvalues $(\gamma'_j)_{j \geq 1}$, $(\lambda'_j)_{j \geq 1}$. Similarly, define C''_y, K''_y by replacing $\boldsymbol{y}^*$ with $(\boldsymbol{y}^*, y)$. We make explicit the dependence on y by writing $(\lambda''_j(y))_{j \geq 1}$. Following Example 3.11, the ratio of expected values in Theorem 3.3 equals*

$$\exp\left\{\log \psi(u) \left(\sum_{j \geq 1} \lambda''_j(y) - \lambda'_j\right)\right\}.$$

Moreover, by the Schur determinant identity, denoting by $K_{y,y} := K(y,y)$, $K_{\boldsymbol{y}^*,y} := (K(y_1^*,y),\dots,K(y_k^*,y))^\top$, and by $K_{\boldsymbol{y}^*,\boldsymbol{y}^*}$ the $k \times k$ matrix with entries $(K(y_i^*,y_j^*))_{i,j=1}^k$ we have

$$\frac{m_{\Phi^{k+1}}(\boldsymbol{y}^*,y)}{m_{\Phi^k}(\boldsymbol{y}^*)} P_0(\mathrm{d}y) = \left(K_{y,y} - K_{\boldsymbol{y}^*,y}^\top K_{\boldsymbol{y}^*,\boldsymbol{y}^*}^{-1} K_{\boldsymbol{y}^*,y}\right)\mathrm{d}y$$

3.4.3 Distribution of the distinct values

From the marginal characterization in Theorem 3.2, it is easy to derive the joint distribution of $(K_n, \boldsymbol{Y}_n^*)$, that is, the joint distribution of the number of the distinct values and their position in a sample of size n.

Proposition 3.2. *Given a set of distinct points $\boldsymbol{y}^* = (y_1^*,\dots,y_k^*)$, let $(q_r)_{r\geq 0}$ be the probability mass function of the number of points in $\Phi^!_{\boldsymbol{y}^*}$, i.e., $q_r := \mathsf{P}\left(\Phi^!_{\boldsymbol{y}^*}(\mathbb{X}) = r\right)$. Define*

$$V(n_1,\dots,n_k;r) := \int_{\mathbb{R}_+} \frac{u^{n-1}}{\Gamma(n)} \psi(u)^r \prod_{j=1}^k \kappa(u,n_j)\mathrm{d}u,$$

then the joint distribution of K_n and $\boldsymbol{Y}^*$ equals

$$\mathsf{P}(K_n = k, \boldsymbol{Y}^* \in \mathrm{d}\boldsymbol{y}^*) = \frac{1}{k!}\sum_{r=0}^{\infty}\left(\sum_{n_1+\dots+n_k=n}\binom{n}{n_1 \cdots n_k} V(n_1,\dots,n_k;r)\right) q_r \; M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*).$$

It is interesting to specialize Proposition 3.2 for a special choice of the distribution of the jumps S_j.

Corollary 3.1. *Under the same assumptions of Proposition 3.2 and by assuming that the S_j's are Gamma$(\alpha, 1)$ distributed,*

$$\mathsf{P}(K_n = k, \boldsymbol{Y}^* \in \mathrm{d}\boldsymbol{y}^*) = \frac{1}{\Gamma(n)}\alpha^k S_{n,k}^{-1,k}\left(\sum_{r\geq 0} q_r \frac{\Gamma\,(k+r)\alpha)}{\Gamma\,((k+r)\alpha+n)}\right) M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*) \qquad (3.15)$$

where $\Gamma(\cdot)$ is the gamma function and $S_{n,k}^{-1,k}$ denotes the generalized Stirling number.

Using Corollary 3.1, we consider now a concrete example highlighting the difference between a repulsive and a non-repulsive point process to show the great flexibility of our prior with respect to traditional ones based on Poisson processes. To this end, we computed $\mathsf{P}(K_n = k, \boldsymbol{Y}^* \in \mathrm{d}\boldsymbol{y}^*)$ for $n = 5$ and Gamma$(1,1)$ distributed weights under two possible priors for Φ: a Poisson process and a DDP. In particular, the Poisson process prior has intensity $\nu(\mathrm{d}x) = \mathbb{1}_R(\mathrm{d}x)$ where $R = (-1/2, 1/2)$. The DPP prior is defined on R as well and is characterized by a Gaussian covariance function $K(x,y) = 5\exp(-(x-y)^2/0.3)$. We consider different settings: in the first (I) $\boldsymbol{y}^* = (-x, x)$, in the second one (II) $\boldsymbol{y}^* = (-0.3, -0.3+2x)$ and in the third (III) $\boldsymbol{y}^* = (-x, 0, x)$. Moreover, we assume that x varies in the interval $(0, 0.4)$. Figure 3.4.1 shows the joint probability of K_n and $\boldsymbol{y}^*$ under the different scenarios. Note that under the Poisson process prior (solid line, left plot), the probability does not depend on $\boldsymbol{y}^*$. Instead, under the DPP prior, the probability increases when the points in $\boldsymbol{y}^*$ are well separated. Note that the values of the different probabilities $\mathsf{P}(K_n = k, \boldsymbol{Y}^* \in \mathrm{d}\boldsymbol{y}^*)$ in Figure 3.4.1 are immaterial, but their behaviors highlight the flexibility of the new class of priors which allow to create well-separated clusters.

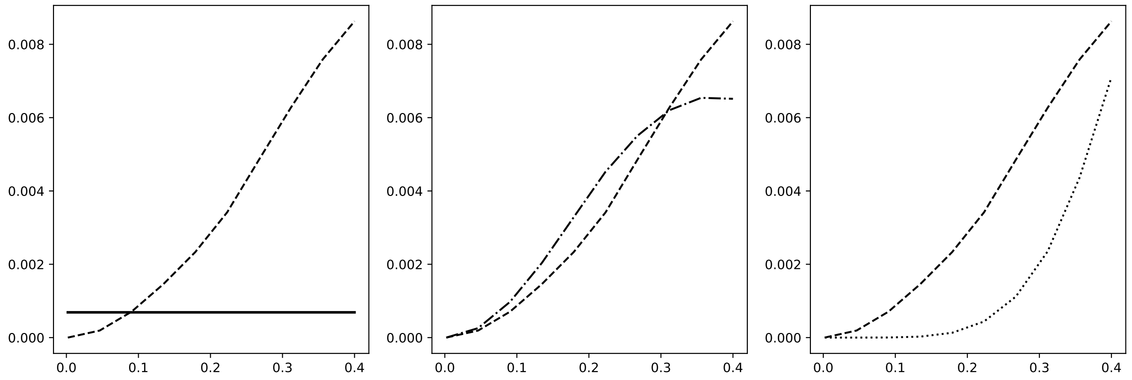


Figure 3.4.1: $\mathsf{P}(K_n = k, \boldsymbol{Y}^* \in \mathrm{d}\boldsymbol{y}^*)$ when $n = 5$, $\alpha = 1$, under different settings. Left plot, setting (I) under the Poisson process (——) and DPP (- - -) prior. Middle plot: setting (I) (- - -) and setting (II) (-·-·) under the DPP prior. Right plot setting (I) (- - -) and setting (III) (······) under the DPP prior.

3.5 Bayesian Hierarchical Mixture Models

Discrete random probability measures are commonly employed as prior for mixing measures in Bayesian model-based clustering. In a mixture model, instead of modeling observations through (3.2), we use it as a prior for latent variables $Y_1, \ldots, Y_n$. Then, for $\mathbb{Z}$-valued observations $Z_1, \ldots, Z_n$, we assume that $Z_i \mid Y_i = y \overset{\text{ind}}{\sim} f(\cdot \mid y)$ where f is a parametric density kernel.

To recover traditionally employed location-scale mixtures, it is convenient to consider random measures on an extended space $\mathbb{X} \times \mathbb{W}$ so that

$$\widetilde{\mu}(\cdot) = \sum_j S_j \delta_{(X_j, W_j)} \tag{3.16}$$

obtained by marking the points in the point process Ψ with i.i.d. marks $(W_j)_{j \geq 1}$ from an absolutely continuous probability distribution over $\mathbb{W}$, whose density we will denote by $f_W(\cdot)$.

To formalize the mixture model, consider observations $Z_1, \ldots, Z_n \in \mathbb{Z}$ and a probability kernel $f : \mathbb{Z} \times \mathbb{X} \times \mathbb{W} \to \mathbb{R}_+$, such that $z \mapsto f(z \mid y, v)$ is a probability density over $\mathbb{Z}$ for any $y, v \in \mathbb{X} \times \mathbb{W}$. We assume

$$\begin{aligned} Z_i \mid Y_i, V_i &\overset{\text{ind}}{\sim} f(\cdot \mid Y_i, V_i), \qquad i = 1, \ldots, n \\ Y_i, V_i \mid \widetilde{\mu} &\overset{\text{iid}}{\sim} \frac{\widetilde{\mu}}{\widetilde{\mu}(\mathbb{X} \times \mathbb{W})} \\ \widetilde{\mu} &\sim \mathbf{P}_\mu \end{aligned} \tag{3.17}$$

Usually, the kernel $f(\cdot \mid Y_i, V_i)$ is the Gaussian distribution with mean Y_i and variance (if data are univariate) or covariance matrix (if data are multivariate) V_i. Hence, the points X_j are the component-specific means and the points W_j the component-specific variances in a Gaussian mixture model.

For posterior simulation, it is convenient to introduce auxiliary cluster indicator variables C_i, $i = 1, \ldots, n$ such that $C_i \mid \widetilde{\mu}$ is a discrete random variable with support $\{1, \ldots, m\}$ such that $\mathsf{P}(C_i = h \mid \widetilde{\mu}) \propto S_h$. Then, observe that $(Y_i, V_i)_{i \geq 1} = (X_{C_i}, W_{C_i})_{i \geq 1}$ so that (3.17) is equivalent to

$$\begin{aligned} Z_i \mid \widetilde{\mu}, C_i &\overset{\text{ind}}{\sim} f(\cdot \mid X_{C_i}, W_{C_i}), \qquad i = 1, \dots, n \\ C_i \mid \widetilde{\mu} &\overset{\text{iid}}{\sim} \text{Categorical}(S_1/T, \dots, S_m/T) \\ \widetilde{\mu} &\sim \mathbf{P}_\mu \end{aligned} \tag{3.18}$$

where $T := \sum_{j \geq 1} S_j$. Therefore, the "distinct values" $(Y_1^*, V_1^*) \dots (Y_{K_n}^*, V_{K_n}^*)$ in the sample are now represented by the distinct values in $(X_{C_i}, W_{C_i})_{i \geq 1}$. Following the standard terminology in Bayesian mixture analysis (Argiento and De Iorio, 2022; Griffin and Walker, 2011), we define $\boldsymbol{S}^{(a)}$, $\boldsymbol{S}^{(a)} = (S_1^{(a)}, \dots, S_{K_n}^{(a)})$ as the distinct values in $(S_{C_i})_{i \geq 1}$ and refer to them as *active* jumps. Moreover, we set the *non-active* jumps $\boldsymbol{S}^{(na)} := \{S_1, \dots, S_m\} \setminus \boldsymbol{S}^{(a)}$. In an analogous way, we define $\boldsymbol{X}^{(a)}, \boldsymbol{X}^{(na)}$ and $\boldsymbol{W}^{(a)}, \boldsymbol{W}^{(na)}$ and refer to them as the active and non-active atoms, respectively.

The posterior distribution of $\widetilde{\mu} \mid Z_1, \dots, Z_n$ is not available in closed form. In the rest of this section, we describe two Markov chain Monte Carlo algorithms for posterior inference in model (3.17). Following Papaspiliopoulos and Roberts (2008) we term them *conditional* and *marginal* respectively. In the conditional one, the random measure $\widetilde{\mu}$ is part of the state of the algorithm, while in the marginal one, it is integrated out. We will assume that, conditionally to the number of points and other eventual hyperparameters, the distribution ordered points $(X_1, \dots, X_M) \mid M = m$ in Φ admits a density with respect to a m-fold product measure defined on $\mathbb{X}^m$, usually the Lebesgue one. With abuse of notation, we denote this density with f_Φ (observe that, indeed, f_Φ is proportional to the point process density with respect to the unit rate Poisson point process).

3.5.1 A conditional MCMC algorithm

Theorem 3.1 can be used to derive the full-conditional of $\widetilde{\mu}$ given $C_1, \dots, C_n, \boldsymbol{X}^{(a)}, \boldsymbol{W}^{(a)}$ and U. This needs a trivial extension to encompass for i.i.d. marks $(W_j)_{j \geq 1}$ defining the support of $\widetilde{\mu}$.

Corollary 3.2. *Consider model* (3.18). *Let* $n_h = \sum_{i=1}^n \mathbb{1}_h(C_i)$. *Then, conditionally* $C_1, \dots, C_n, \boldsymbol{X}^{(a)} = \boldsymbol{x}^{(a)}, \boldsymbol{W}^{(\boldsymbol{a})} = \boldsymbol{w}^{(a)}$ *and* $U = u$, $\widetilde{\mu}$ *is distributed as*

$$\sum_{h=1}^k S_h^{(a)} \delta_{(x_h^{(a)}, w_h^{(a)})} + \widetilde{\mu}'$$

where $S_h^{(a)} \sim f_{S_h^{(a)}}(s) \propto s^{n_h} e^{-us} H(\mathrm{d}s)$, *and* $\widetilde{\mu}' := \sum_{h \geq 1} S_h^{(na)} \delta_{(X_h^{(na)}, W_h^{(na)})}$ *is obtained by considering the random measure at point (ii) in Theorem 3.1 and adding i.i.d. marks* $W_h^{(na)} \sim f_W$ *to the support points.*

This suggests the following algorithm, where we denote by "$\cdot \mid \text{rest}$" conditioning with respect all the variables not appearing on the left hand side of the conditioning symbol.

1. Sample $U \mid \text{rest} \sim \text{Gamma}(n, T)$, where $T := \sum_{j=1}^m S_j$

2. Sample each C_i independently from a discrete distribution over $\{1, \dots, m\}$ such that

$$\mathsf{P}(C_i = h \mid \text{rest}) \propto S_h f(Z_i \mid X_h, W_h)$$

Set k equal to the cardinality of the unique values in $\boldsymbol{X}^a := \{X_{C_i},\ i = 1, \dots, n\}$. Define $\boldsymbol{W}^a$ and $\boldsymbol{S}^a$ analogously Then, relabel $\boldsymbol{S}$, $\boldsymbol{X}$ and $\boldsymbol{W}$ so that $X_1, \dots, X_k$ equals $\boldsymbol{X}^a$ and analogously for $\boldsymbol{S}$, $\boldsymbol{W}$ with respect to $\boldsymbol{S}^a$, $\boldsymbol{W}^a$.

3. Sample $\widetilde{\mu}$ using the distribution in Corollary 3.2. That is, sample S_h^a from a distribution on $\mathbb{R}_+$ with density

$$f_{S_h}(s) \propto e^{-Us} s^{n_h} H(\mathrm{d}s)$$

   and $\widetilde{\mu}' := \sum_{j \geq 1} S_j^{na} \delta_{(X_j^{na}, W_j^{na})}$ the law of the random measure with Laplace transform (3.9).

4. Sample $\boldsymbol{X}^a$ from

$$\mathsf{P}(\boldsymbol{X}^a \in \mathrm{d}\boldsymbol{x}^a \mid \text{rest}) \propto f_\Phi(\boldsymbol{x}^a, \boldsymbol{X}^{na}) \prod_{h=1}^{k} \prod_{i : C_i = h} f(Y_i \mid x_h^a, W_h)$$

5. Sample each entry in $\boldsymbol{W}^a$ independently from

$$\mathsf{P}(W_h^a \in \mathrm{d}w) \propto f_W(w) \prod_{i : C_i = h} f(Y_i \mid X_h^a, w)$$

6. Set $\boldsymbol{X} = (X^a, X^{na})$, $\boldsymbol{W} = (W^a, W^{na})$ $\boldsymbol{S} = (S^a, S^{na})$ and m equal to the length of these vectors.

Of the above steps, the most complex one is surely sampling $\widetilde{\mu}'$. In fact, all the remaining ones can be handled either by closed form full-conditionals (depending for instance on the law of the jumps $H(\mathrm{d}s)$ and the prior f_W) or by simple Metropolis-Hastings steps. When Φ is a Gibbs point process with a density, we recover the same algorithm in Beraha et al. (2022), where the update of $\widetilde{\mu}'$ was performed by sampling $\boldsymbol{X}^{(na)}$ from the law of a point process via a birth-and-death Metropolis-Hastings algorithm. Our understanding of the posterior distribution in Theorem 3.1 allows for tailored algorithms to specific cases where the distribution of $\boldsymbol{X}^{(na)}$ is known. This is for instance the case of a DPP prior. In this case, instead of employing a Metropolis-Hastings step for $\boldsymbol{X}^{(na)}$, we can use Algorithm 1 in Lavancier et al. (2015) to obtain a perfect sample from the law of $\boldsymbol{X}^{(na)}$. We expect this choice to yield superior performance in terms of mixing.

3.5.2 A marginal MCMC algorithm

When integrating (3.17) with respect to $\widetilde{\mu}$, we can exploit Theorem 3.3 to devise a marginal MCMC strategy. Using the generalized restaurant Chinese restaurant metaphor, at every iteration of the MCMC algorithm, each customer is removed from the restaurant and reenters following the conditional distribution in Theorem 3.3. The plain application of this result yields the following MCMC algorithm

1. Sample $U \mid \text{rest}$ from the full conditional distribution in (3.10)

2. For each observation, sample (Y_i, V_i) from

$$\mathsf{P}(Y_i, V_i \in \mathrm{d}y\,\mathrm{d}v \mid \text{rest}) \propto \sum_{j=1}^{k} \frac{\kappa(u, n_j^{(-i)} + 1)}{\kappa(u, n_j^{(-i)})} f(Z_i \mid Y_j^*, V_j^*) \delta_{(Y_j^*, V_j^*)}(\mathrm{d}y\,\mathrm{d}v) +$$
$$\kappa(u, 1) \frac{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u \widetilde{\mu}^!_{k+1]}(\mathrm{d}z)}\right]}{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u \widetilde{\mu}^!_{k}(\mathrm{d}z)}\right]} \int_{\mathbb{X} \times \mathbb{W}} \frac{m_{\Phi^{k+1}}(\boldsymbol{y}^{*(-i)}, y)}{m_{\Phi^k}(\boldsymbol{y}^{*(-i)})} f(Z_i \mid y, v) P_0(\mathrm{d}y) f_W(v) \mathrm{d}v.$$

   where the superscript $(-i)$ means that the i-th observation is removed from the state for the computations. Here, $\boldsymbol{Y}^*$ is the unique values in $(Y1, \ldots, Y_n)$ and similarly for $\boldsymbol{V}^*$.

3. Sample the unique values $\boldsymbol{Y}^*$ from a joint distribution proportional to

$$f_\Phi(\boldsymbol{y}) \prod_{h=1}^{k} \prod_{i : Y_i = Y^*_h} f(Z_i \mid y_h, V^*_h)$$

4. Sample each of unique values $\boldsymbol{V}^*$ independently from

$$\mathsf{P}(V^*_h \in \mathrm{d}v) \propto f_W(v) \prod_{i : C_i = h} f(Y_i \mid Y^*_h, v)$$

Step (2) of the algorithm above requires the computation of

$$\int_{\mathbb{X} \times \mathbb{W}} \frac{m_{\Phi^{k+1}}(\boldsymbol{y}^{*(-i)}, y)}{m_{\Phi^k}(\boldsymbol{y}^{*(-i)})} f(Z_i \mid y, v) P_0(\mathrm{d}y) f_W(v) \mathrm{d}v$$

which might be challenging is situations where data are multidimensional. Note that an analogous challenge is faced by the algorithm proposed by Xie and Xu (2019), where the authors use numerical quadrature techniques to evaluate a similar integral. Our approach has the advantage that is more general (they focus only on a specific class of Gibbs point processes, namely pairwise interaction point processes) and does not require the numerical evaluation of the normalizing constant of the point process density. Moreover, in higher dimensional setting we can adapt the strategy devised by Neal in his Algorithm 8, where we introduce L auxiliary variables and replace step (2) with

2'.a for $\ell = 1, \ldots, L$, sample $Y^*_{k+\ell}$ from

$$\mathsf{P}(Y^*_{k+\ell} \in \mathrm{d}y \mid \mathrm{rest}) \propto \frac{m_{\Phi^{k+1}}(\boldsymbol{y}^{*(-i)}, y)}{m_{\Phi^k}(\boldsymbol{y}^{*(-i)})} P_0(\mathrm{d}y)$$

and $V^*_{k+\ell} \overset{\mathrm{iid}}{\sim} f_W$.

2'.b Set (Y_i, V_i) equal to (Y^*_h, V^*_h) with probability proportional to

$$\begin{aligned} &\frac{\kappa(u, n_j^{(-i)} + 1)}{\kappa(u, n_j^{(-i)})} f(Z_i \mid Y^*_h, V^*_h), \qquad h = 1, \ldots, k \\ &\frac{1}{L} \kappa(u, 1) \frac{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u \widetilde{\mu}^!_{k+1}(\mathrm{d}z)}\right]}{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u \widetilde{\mu}^!_{k}(\mathrm{d}z)}\right]} f(Z_i \mid Y^*_h, V^*_h), \qquad h = k+1, \ldots, k+L. \end{aligned}$$

3.6 Shot-Noise Cox Process Mixtures

Cox processes (Cox, 1955), also known as doubly stochastic Poisson processes, can be defined via the hierarchical model

$$\begin{aligned} \Phi \mid \Lambda &\sim \mathrm{PRM}(\nu_\Lambda) \\ \Lambda &\sim \mathcal{P}_\Lambda \end{aligned} \tag{3.19}$$

where $\mathrm{PRM}(\nu_\Lambda)$ denotes the law of a Poisson random measure with intensity $\nu_\Lambda(x)\mathrm{d}x$. We consider here a special case of (3.19) termed the shot-noise Cox process, introduced in Møller (2003), for which $\Lambda \sim \mathrm{PRM}(\rho)$, for some nonatomic sigma-finite measure $\nu(x)\mathrm{d}x$ on $\mathbb{X}$ and

$$\nu_\Lambda(x) := \gamma \int_{\mathbb{X}} k_\alpha(x - v) \Lambda(\mathrm{d}v).$$

where k_α is a probability density and $\gamma > 0$. In particular, by assuming that $x \mapsto k_\alpha(x-v)$ is continuous for any v, we get that the point process Φ is simple.

We will refer to ρ as the base intensity of the process Φ. It is easy to see that $M_\Phi(A) = \int_A m_\Phi(x)\mathrm{d}x = \gamma \int_A \int_{\mathbb{X}} k_\alpha(x-v)\nu(\mathrm{d}v)\mathrm{d}x$. See Lemma 3.6 and Lemma 3.4 in the appendix for the n-th moment measure and the Palm version of shot noise cox processes respectively

The following result specializes Theorem 3.1 in the case of shot-noise Cox processes.

Theorem 3.4. *Assume that Φ is a shot-noise Cox process with base intensity $\nu(\mathrm{d}x)$. Then, the random measure $\widetilde{\mu}'$ in (3.8) can be decomposed as*

$$\widetilde{\mu}' = \widetilde{\mu}_0 + \sum_{h=1}^{k} \widetilde{\mu}_{\zeta_h},$$

where $\widetilde{\mu}_0 = \sum_{j\geq 1} \tilde{S}_j \delta_{\tilde{X}_j}$ and $\widetilde{\mu}_{\zeta_h} = \sum_{j \geq 1} \tilde{S}_{h,j}\delta_{\tilde{X}_{h,j}}$. The unmarked point processes $\Phi_0 = \sum_{j\geq 1} \delta_{\tilde{X}_j}$ is a shot-noise Cox process with base intensity

$$\rho'(\mathrm{d}x) = e^{-\gamma(1-\psi(u))}\nu(\mathrm{d}x),$$

each $\Phi_{\zeta_h} = \sum_{j\geq 1} \delta_{\tilde{X}_{h,j}}$ is a Point process with random intensity

$$\nu_{\zeta_h} = e^{-\gamma(1-\psi(u))}k_\alpha(\zeta_h - x)\mathrm{d}x,$$

and the random variables ζ_h are as in Proposition 3.4. Finally, all the weights of $\widetilde{\mu}_0$ and the $\widetilde{\mu}_{\zeta_h}$'s are i.i.d. with distribution

$$H'(\mathrm{d}s) := \frac{e^{-su}H(\mathrm{d}s)}{\int_{\mathbb{R}^+} e^{-su}H(\mathrm{d}s)}.$$

For simplicity, hereafter we assume that both the base measure $\nu(\mathrm{d}x)$ and the kernel k_α integrate to 1. The following results are trivial to extend to other settings.

Proposition 3.3. *Define $\eta(x_1, \ldots, x_l) = \int \prod_{i=1}^{l} k_\alpha(x_i - v)\nu(\mathrm{d}v)$. The marginal distribution of $\boldsymbol{Y}$ under model (3.2) when $\mathbf{P}_\Phi$ is the shot-noise Cox process is*

$$\mathsf{P}(\boldsymbol{Y} \in \mathrm{d}y) = \int_{\mathbb{R}_+} \frac{u^{n-1}}{\Gamma(n)} \exp\left\{\lambda\left(e^{\gamma(\psi(u)^{-1}-1)} - 1\right) + k\gamma(\psi(u)^{-1}-1)\right\} \prod_{j=1}^{k} k(u, n_j)\mathrm{d}u$$
$$\times\, \gamma^k \sum_{j=1}^{k} \sum_{C_1,\ldots C_j \in (*)} \prod_{l=1}^{j} \eta(\boldsymbol{x}_{C_l})\mathrm{d}x_1 \cdots \mathrm{d}x_k.$$

where $(*)$ denotes all the partition of k elements in j groups.

From Corollary 3.1 and Proposition 3.3 it is trivial to derive the marginal distribution of the distinct values. Indeed, from Proposition 3.4 in the Appendix, we have that $\Phi^!_{\boldsymbol{y}}(\mathbb{X}) \mid \Lambda \sim \mathrm{Poi}(\gamma(k + \Lambda(\mathbb{X}))$ and $\Lambda(\mathbb{X}) \sim \mathrm{Poi}(\lambda)$. Hence, $q_r =: \mathsf{P}(\Phi^!_{\boldsymbol{y}}(\mathbb{X}) = r)$ does not depend on the values in $\boldsymbol{y}^*$, but only on the cardinality. Hence, we can marginalize with respect to $\boldsymbol{y}^*$ in Corollary 3.1 observing that the moment measure of Φ is

$$M^k_\Phi(\mathrm{d}\boldsymbol{x}) = \gamma^k \sum_{j=1}^{k} \sum_{C_1,\ldots C_j \in (*)} \prod_{l=1}^{j} \eta(\boldsymbol{x}_{C_l})\mathrm{d}x_1 \cdots \mathrm{d}x_k.$$

See Lemma 3.6 in the Appendix for a proof. Hence, in the case of Gamma$(\alpha, 1)$ distributed jumps

$$\mathsf{P}(K_n = k) = \frac{1}{\Gamma(n)}\alpha^k \gamma^k S_{n,k}^{-1,k} B_k \sum_{r \geq 0} q_r \frac{\Gamma(k+r)\alpha}{\Gamma((k+r)\alpha + n)}.$$

where B_k is the k-th Bell number

Finally, the predictive distribution under the shot-noise Cox process model equals

$$\begin{aligned}\mathsf{P}(Y_{n+1} \in A \mid \boldsymbol{Y} = \boldsymbol{y}, U_n) \propto \sum_{j=1}^{k} \frac{\kappa(U_n, n_j + 1)}{\kappa(U_n, n_j)} \delta_{Y_j^*}(A) \\ + \int_A \kappa(U_n, 1) \exp\left\{\gamma(\psi(u)^{-1} - 1)\right\} \frac{m_{\Phi^{k+1}}(\boldsymbol{y}^*, y)}{m_{\Phi^k}(\boldsymbol{y}^*)} P_0(\mathrm{d}y).\end{aligned}$$

3.6.1 SNCP mixtures as mixtures of mixtures

By the coloring theorem for Poisson point processes (Kingman, 1992), $\Phi \mid \Lambda = \sum_{\lambda_j \in \Lambda} \Phi_j$, where $\Phi_j \mid \Lambda$ is a Poisson point process with intensity $\gamma k_\alpha(x - \lambda_j)$. This shows that SNCPs are *cluster processes*, since for appropriate choices of k_α, the points in a *cluster* Φ_j will be closer than points belonging to different clusters, say Φ_j, Φ_j. When we embed a random probability measure built from (3.19) in a Bayesian mixture model as done in Section 3.5, we obtain that the atoms of the mixture are randomly clustered together. Hence, we can rewrite the mixture density as follows

$$f(z) = \frac{1}{T}\sum_{h \geq 1} \tilde{S}_h f(z \mid \tilde{X}_h, \tilde{W}_h) \equiv \frac{1}{T}\sum_{j=1}^{n(\Lambda)} \sum_{h \geq 1} \tilde{S}_{j,h} f(z \mid \tilde{X}_{j,h}, \tilde{W}_{j,h})$$

where, on the right hand side, we first sum over the atoms of Λ and then in the atoms of each of the Φ_j's. With an abuse of notation, we introduced a second subscript to $\tilde{S}_h$, $\tilde{X}_h$, and $\tilde{W}_h$ to represent that each point of $\tilde{X}_h$ (and its marks) can be assigned to a point in Λ. Let now

$$\tilde{F}_j(z) = \frac{1}{P_j}\sum_{h \geq 1} \tilde{S}_{j,h} f(z \mid \tilde{X}_{j,h}, \tilde{W}_{j,h}), \qquad P_j := \sum_{h \geq 1} \tilde{S}_{j,h} \tag{3.20}$$

which is a (random) probability density function on $\mathbb{X}$. We clearly have $f(z) = T^{-1}\sum_{j=1}^{n(\Lambda)} P_j \tilde{F}_j(z)$. Therefore, an SNCP mixture model can be written as a *mixture of mixtures*, where each component $\tilde{F}_j(z)$ is expressed as a mixture model itself. We can thus regard the SNCP mixture model as a nonparametric generalization of the model in Malsiner-Walli et al. (2017). In the frequentist setting, identifiability and estimation of the *mixture of mixtures* model have recently been studied in Aragam et al. (2020).

3.6.2 Numerical Illustration

We consider a simulated scenario where 100 datapoints are generated from a two-component mixture of Student's t distribution with 3 degrees of freedom, centered respectively in -5 and $+5$. We fit to the dataset a mixture of Gaussian distributions, so that $f(\cdot \mid \cdot)$ in (3.17) is the Gaussian density with parameters (y_i, τ_i) representing mean and variance, respectively. As prior for the γ_j's we assume an inverse-Gamma distribution with shape and scale parameter equal to two. The prior for Φ is the shot-noise Cox process where we set $\alpha = 1$ and $\gamma = 1$. Finally, the unnormalized weights s_j are given a Gamma prior with shape and scale equal to two.

We compare the SNCP mixture to the finite mixture model in Argiento and De Iorio (2022), where the mixing measure μ is as in Equation (3.16). The number of support

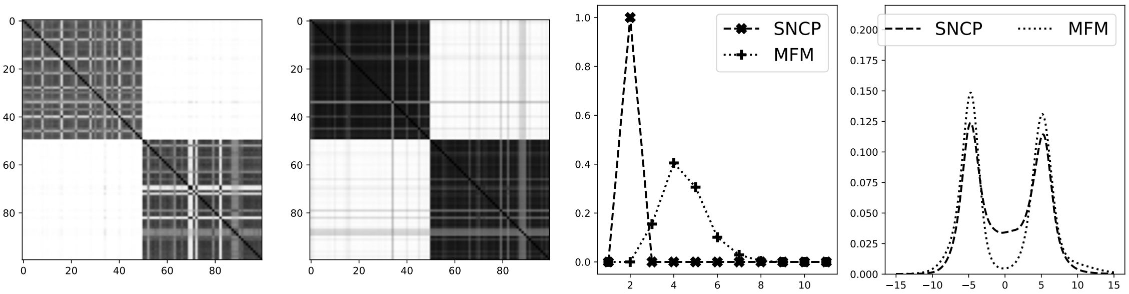


Figure 3.6.1: From left to right: posterior similarity matrix of the cluster labels under MFM and SNCP mixture, posterior distribution of the number of clusters, density estimate.

points K is given a shifted Poisson distribution, so that $K - 1 \sim \text{Poi}(2)$. Given K, the support points (x_j, γ_j) are assumed i.i.d. from a normal-inverse-gamma distribution. The unnormalized weights s_j are given the same prior as for the SNCP mixture model. Posterior inference is computed via the `BayesMix` library.

To fit the SNCP mixture, we use the conditional algorithm in Section 3.5.1, where we further add to the MCMC state the points of Λ, see Section 3.G.4 for further details

The SNCP mixture allocates between 25 and 40 Gaussian components to represent the mixture of Student t's distribution, while the MFM model between 3 and 8. To cluster data under the SNCP model, we do not consider directly the c_i's as cluster indicator labels, but refer each cluster to the associated atom in the directing Poisson process Λ. That is, using notation as in Section 3.G.4, we partition data according to the labels t_{c_i}. Posterior inference is summarized in Figure 3.6.1. It is clear that the SNCP mixutre does a better job in dividing data into two clusters. In particular, the posterior probability of having two clusters under MFM is zero.

3.7 Discussion

In this work, we have provided a general construction for random probability measures with interaction across support points. Our approach is similar in spirit to the construction of NRMIs (Regazzini et al., 2003), but we use general point processes other than the Poisson one to induce the desired dependence. We establish several distributional results, which are useful for prior elicitation as well as for posterior simulation in Bayesian hierarchical mixture models. Our general theory is illustrated through the examples of Poisson, Gibbs, and determinantal point processes. Furthermore, we discuss the use of shot-noise Cox processes, which were not considered previously in connection with Bayesian mixture models.

We plan to investigate several extensions. First, we could consider a version of shot-noise Cox processes where Λ is not Poisson. For instance, we could assume the determinantal shot-noise Cox process introduced in Møller and Vihrs (2022), which, in principle, would favor well separated random components $\tilde{F}_j$ (see (3.20)). Moreover, we could relax the assumption that Φ is simple (i.e., its atoms are pairwise different with probability one), and consider hierarchical models such as $\Phi \mid \Phi_0 \sim \text{PRM}(\Phi_0)$ and $\Phi_0 \sim \mathbf{P}$, where $\mathbf{P}$ is a general law for a point process. This construction could be extended to the *partially exchangeable* by considering a counterpart of the hierarchical processes in Camerlenghi et al. (2019). For instance, we could assume $\Phi_i \mid \Phi_0 \overset{\text{iid}}{\sim} \text{PRM}(\Phi_0)$, $i = 1, \ldots, g$ where each random measure Φ_i is used to model a different collection of observations and the hierarchical model for the Φ_i's yields a "borrowing of strength" that is typical of Bayesian inference. Along these lines, it would also be interesting to consider an extension of the (determinantal) shot-noise Cox process mixture to the partially exchangeable case, by letting

$$\Phi_i \mid \Lambda \overset{\text{iid}}{\sim} \text{PRM}\left(\gamma \int k_\alpha(\cdot - v)\Lambda(\mathrm{d}v)\right), \qquad i = 1, \ldots, g.$$

Then, we could cluster observations (both within and across groups) based on the atoms of Λ, which could result in a more flexible cluster detection. Finally, we could consider more general classes of models such as feature sampling models or trait allocation models.

Another important aspect we plan to work on, is proposing more efficient MCMC algorithms, for instance similar to the split-merge algorithm in Jain and Neal (2004) or based on variational inference (Blei and Jordan, 2006) to handle moderately-dimensional observations and parametric spaces.

Appendix

3.A Background on Palm calculus

The basic tool used in our computations is a disintegration of the Campbell measure of a point process Φ with respect to its mean measure M_Φ, usually called Palm kernel or family of Palm distributions of Φ. Below, we recall the main results needed later in this chapter. For further details about Palm distributions and Palm calculus see, e.g., the papers Kallenberg (1984); Coeurjolly et al. (2017) or the monographs Kallenberg (2017) (Chapter 6), Daley and Vere-Jones (2008) (Chapter 13). Here, we adapt the notation from the recent monograph Baccelli et al. (2020) (Chapter 3).

Let $\mathbb{M}(\mathbb{X})$ the space of bounded measures on $\mathbb{X}$ and denote with $\mathcal{M}(\mathbb{X})$ its σ-algebra. For a point process Φ on $\mathbb{X}$, let the mean measure M_Φ: $M_\Phi(B) := \mathbb{E}[\Phi(B)]$ for all $B \in \mathcal{X}$. We define the Campbell measure $\mathcal{C}_\phi$ on $\mathbb{X} \times \mathcal{M}(\mathbb{X})$ as

$$\mathcal{C}_\Phi(B \times L) = \mathbb{E}\left[\Phi(B)\mathbb{1}[\Phi \in L]\right], \quad B \in \mathcal{X}, L \in \mathcal{M}(\mathbb{X}).$$

Then, there exists a M_Φ-a.e. unique disintegration probability kernel $\{\mathbf{P}_\Phi^x(\cdot)\}_{x\in\mathbb{X}}$ of $\mathcal{C}_\phi$ with respect to M_Φ, i.e.

$$\mathcal{C}_\Phi(B \times L) = \int_B \mathbf{P}_\Phi^x(L) M_\Phi(\mathrm{d}x) \quad B \in \mathcal{X}, L \in \mathcal{M}(\mathbb{X})$$

Note that, for any $x \in \mathbb{X}$, $\mathbf{P}_\Phi^x$ is the distribution of a random measure (specifically, a point process) on $\mathbb{X}$. Therefore, $\mathbf{P}_\Phi^x$ can be identified with the distribution of point process Φ_x such that $\mathbf{P}_\Phi^x(L) = \mathsf{P}(\Phi_x \in L)$. Following Baccelli et al. (2020) we call Φ_x the Palm version of Φ at x.

Theorem 3.5 (Campbell-Little-Mecke formula. Theorem 3.1.9 in Baccelli et al. (2020))**.** *Let Φ a point process on $\mathbb{X}$ such that M_Φ is σ-finite. Denote with $\mathbf{P}_\Phi(\cdot)$ its law. Let $\{\mathbf{P}_\Phi^x(\cdot)\}_{x\in\mathbb{X}}$ a family of Palm distributions of Φ. Then, for all measurable $g : \mathbb{X} \times \mathbb{M}(\mathbb{X}) \to \mathbb{R}_+$*

$$\mathbb{E}\left[\int_{\mathbb{X}} g(x, \Phi)\Phi(\mathrm{d}x)\right] = \int_{\mathbb{M}(\mathbb{X})\times\mathbb{X}} g(x,\nu)\nu(\mathrm{d}x)\mathbf{P}_\Phi(\mathrm{d}\nu) = \int_{\mathbb{X}\times\mathbb{M}(\mathbb{X})} g(x,\nu)\mathbf{P}_\Phi^x(\mathrm{d}\nu)\mathbb{M}_\Phi(\mathrm{d}x) \tag{3.21}$$

In many applications in spatial statistics, the Campbell-Little-Mecke formula is stated in terms of the *reduced* Palm kernel $\mathbf{P}_{\Phi^!}^x$

$$\mathbb{E}\left[\int_{\mathbb{X}} g(x, \Phi - \delta_x)\Phi(\mathrm{d}x)\right] = \int_{\mathbb{X}\times\mathbb{M}(\mathbb{X})} g(x,\nu)\mathbf{P}_{\Phi^!}^x(\mathrm{d}\nu)\mathbb{M}_\Phi(\mathrm{d}x)$$

where $\Phi - \delta_x$ is obtained by removing the point x from Φ and $\mathbf{P}_{\Phi^!}^x$ is the distribution of the point process

$$\Phi_x^! := \Phi_x - \delta_x.$$

Hence, given a reduced Palm kernel, we can construct the nonreduced one by considering the distribution of $\Phi^!_x + \delta_x$.

Given a point process Φ define the n-th Campbell measure

$$\mathcal{C}^n_\Phi(B \times L) = \mathbb{E}\left[\int_B \mathbb{1}[\Phi \in L]\Phi^n(\mathrm{d}\boldsymbol{x})\right], \quad B \in \mathcal{X}^{\otimes n}, L \in \mathcal{M}(\mathbb{X})$$

where $\mathrm{d}\boldsymbol{x} = \mathrm{d}x_1 \cdots \mathrm{d}x_n$ and $\Phi^n(\mathrm{d}\boldsymbol{x}) = \prod_{i=1}^n \Phi(\mathrm{d}x_i)$. Let M^n_Φ be the mean measure of Φ^n, i.e. $M_{\Phi^n}(B) = \mathcal{C}^N(B \times \mathcal{M}(\mathbb{X}))$, then the n–th Palm distribution $\{\mathbf{P}^{\boldsymbol{x}}_\Phi\}_{\boldsymbol{x}\in\mathbb{X}^n}$ is defined as the disintegration kernel of C^n with respect to M_{Φ^n}, that is

$$\mathcal{C}^n_\Phi(B \times L) = \int_B \mathbf{P}^{\boldsymbol{x}}_\Phi(L) M_{\Phi^n}(\mathrm{d}\boldsymbol{x}), \quad B \in \mathcal{X}^{\otimes n}, L \in \mathcal{M}(\mathbb{X})$$

The following is a multivariate extension to Theorem 3.5 that will be useful for later computations

Theorem 3.6 (Higher order CLM formula.). *Let Φ a point process on $\mathbb{X}$ such that M_{Φ^n} is σ-finite. Let $\{\mathbf{P}^{\mathbf{x}}_\Phi(\cdot)\}_{\mathbf{x}\in\mathbb{X}^n}$ a family of n–th Palm distributions of Φ. Then, for all measurable $g : \mathbb{X}^n \times \mathbb{M}(\mathbb{X}) \to \mathbb{R}_+$*

$$\mathbb{E}\left[\int_{\mathbb{X}^n} g(\boldsymbol{x}, \Phi)\Phi^n(\mathrm{d}\boldsymbol{x})\right] = \int_{\mathbb{X}^n \times \mathbb{M}(\mathbb{X})} g(\boldsymbol{x}, \nu)\mathbf{P}^{\mathbf{x}}_\Phi(\mathrm{d}\nu)\mathbb{M}_{\Phi^n}(\mathrm{d}\boldsymbol{x}) \tag{3.22}$$

3.B Preparatory Lemmas

We now state some results relating to independently marked point processes

Lemma 3.1. *Let Ψ be an independently marked point process on $\mathbb{X}\times\mathbb{S}$ obtained by marking a point process Φ on $\mathbb{X}$ with i.i.d marks $\{s_j\}_j$ from a probability measure H, that does not depend on the value of the associated atom x_j. Then, the mean measure $M_\Psi(\mathrm{d}x, \mathrm{d}s)$ is*

$$M_\Psi(\mathrm{d}x, \mathrm{d}s) = H(\mathrm{d}s)M_\Phi(\mathrm{d}x)$$

Analogously, defining $\Psi^n(\mathrm{d}\boldsymbol{x}, \mathrm{d}\boldsymbol{s}) = \prod_{i=1}^n \Psi(\mathrm{d}x_i, \mathrm{d}s_i)$, the n–th mean measure of Ψ equals

$$M_{\Psi^n}(\mathrm{d}\boldsymbol{x}, \mathrm{d}\boldsymbol{s}) = H^n(\mathrm{d}\boldsymbol{s})M_{\Phi^n}(\mathrm{d}\boldsymbol{x})$$

Proof. Let $C = A \times B$, $A \in \mathcal{X}$ $B \in \mathcal{S}$

$$\begin{aligned}\mathbb{E}[\Psi(C)] &= \mathbb{E}\left[\sum \mathbb{1}[x_i \in A, s_i \in B]\right] = \mathbb{E}\left[\sum \mathbb{1}[x_i \in A]\mathbb{1}[s_i \in B]\right] \\ &= \sum \mathbb{E}\left[\mathbb{1}[x_i \in A]\mathbb{1}[s_i \in B]\right] = \sum H(B)\mathbb{E}\left[\mathbb{1}[x_i \in A]\right] \\ &= H(B)\mathbb{E}\left[\sum \mathbb{1}[x_i \in A]\right] = H(B)M_\Phi(B)\end{aligned}$$

The proof for the n–th mean measure is achieved following the same steps with $A \in \mathcal{X}^{\otimes n}$ and $B \in \mathcal{S}^{\otimes n}$. □

Lemma 3.2. *Let Ψ as in Proposition 3.1, then the Palm distribution $\{\mathbf{P}^{x,s}_\Psi\}_{x,s\in\mathbb{X}\times\mathbb{S}}$ is the distribution of the point process $\delta_{(x,s)} + \Psi^!_{x,s}$, where $\Psi^!_{x,s}$ is an independently marked point process obtained by marking $\Phi^!_x \sim \mathbf{P}^x_{\Phi^!}$ with i.i.d marks from H. Similarly, let $(\mathbf{x}, \mathbf{s}) = (x_1, \ldots, x_n, s_1, \ldots, s_n)$, the Palm distribution $\{\mathbf{P}^{\mathbf{x},\mathbf{s}}_\Psi\}_{\mathbf{x},\mathbf{s}\in\mathbb{X}^n\times\mathbb{S}^n}$ is the distribution of the point process $\sum \delta_{(x_i,s_i)} + \Psi^!_{\mathbf{x},\mathbf{s}}$, where $\Psi^!_{\mathbf{x},\mathbf{s}}$ is an independently marked point process obtained by marking $\Phi^!_{\mathbf{x}} \sim \mathbf{P}^{\mathbf{x}}_\Phi$ with i.i.d. marks from H.*

Proof. By the CLM formula for the reduced Palm kernel, we know that $\Psi^!_{x,s}$ satisfies

$$\mathbb{E}\left[\int_{\mathbb{X}\times\mathbb{S}} g(x,s,\Psi-\delta_{(x,s)})\Psi(\mathrm{d}x,\mathrm{d}s)\right] = \int_{\mathbb{X}\times\mathbb{S}} \mathbb{E}\left[g(x,s,\Psi^!_{x,s})\right] M_\Psi(\mathrm{d}x,\mathrm{d}s) \tag{3.23}$$

Consider now the point process $\Psi'_{x,s}$ obtained by marking $\Phi^!_x$ with i.i.d marks from H, if

$$\int_{\mathbb{X}\times\mathbb{S}} \mathbb{E}\left[g(x,s,\Psi'_{x,s})\right] M_\Psi(\mathrm{d}x,\mathrm{d}s) = \int_{\mathbb{X}\times\mathbb{S}} \mathbb{E}\left[g(x,s,\Psi^!_{x,s})\right] M_\Psi(\mathrm{d}x,\mathrm{d}s), \tag{3.24}$$

for any g, we can conclude that $\Psi'_{x,s}$ and $\Psi^!_{x,s}$ are equal in distribution. To prove (3.24), we will show that

$$\int_{\mathbb{X}\times\mathbb{S}} \mathbb{E}\left[g(x,s,\Psi'_{x,s})\right] M_\Psi(\mathrm{d}x,\mathrm{d}s) = \mathbb{E}\left[\int_{\mathbb{X}\times\mathbb{S}} g(x,s,\Psi-\delta_{x,s})\Psi(\mathrm{d}x,\mathrm{d}s)\right]$$

In the following, we indicate with $E_x[f(x,z)]$ that the expectation is taken with respect to the random variable x. Write $\Psi' = (\Phi^!, \boldsymbol{m})$ where $\boldsymbol{m}$ is the collection of marks. With a slight abuse of notation, we write $g(x,s,\Psi'_{x,s}) = g(x,s,\Phi^!_x,\boldsymbol{m})$. Then

$$\begin{aligned}\int_{\mathbb{X}\times\mathbb{S}} \mathbb{E}_\Psi\left[g(x,s,\Psi'_{x,s})\right] M_\Psi(\mathrm{d}x,\mathrm{d}s) &= \int_{\mathbb{X}\times\mathbb{S}} \mathbb{E}_{\Phi^!_x,\mathbf{m}}\left[g(x,s,\Phi^!_x,\boldsymbol{m})\right] M_\Psi(\mathrm{d}x,\mathrm{d}s)\\ &= \int_{\mathbb{X}\times\mathbb{S}} \mathbb{E}_{\Phi^!_x}\left[\mathbb{E}_{\mathbf{m}}\left[g(x,s,\Phi^!_x,\boldsymbol{m}) \mid \Phi^!_x\right]\right] M_\Psi(\mathrm{d}x,\mathrm{d}s)\\ &= \int_{\mathbb{X}\times\mathbb{S}} \mathbb{E}_{\Phi^!_x}\left[\int_{\mathbb{S}^{n^!}} g(x,s,\Phi^!_x,\boldsymbol{m}) \prod_{i:x_i\in\Phi^!_x} H(\mathrm{d}m_i) \mid \Phi^!_x\right] M_\Phi(\mathrm{d}x)H(\mathrm{d}s)\end{aligned}$$

where $n^!$ denotes the cardinality of $\Phi^!_x$. Denoting the cardinality of Φ with n, $n^! = n-1$, By Fubini's theorem, we can interchange the outher most integral over $\mathbb{S}$ with $\mathbb{E}_{\Phi^!_x}$. Then, we apply the CLM formula (in reverse order) with respect to $\Phi^!_x$ obtaining

$$\begin{aligned}&\int_{\mathbb{X}\times\mathbb{S}} \mathbb{E}_\Psi\left[g(x,s,\Psi'_{x,s})\right] M_\Psi(\mathrm{d}x,\mathrm{d}s)\\ &\quad= \mathbb{E}_\Phi\left[\int_{\mathbb{X}}\int_{\mathbb{S}^{n-1+1}} g(x,s,\Phi_x-\delta_x,\boldsymbol{m}) \prod_{i:x_i\in\Phi_x-\delta_x} H(\mathrm{d}m_i)H(\mathrm{d}s)\Phi(\mathrm{d}x)\right]\end{aligned}$$

Now, observe that $(\Phi-\delta_x,\boldsymbol{m}) = \Psi-\delta_{x,s}$ and $\Psi(\mathrm{d}x,\mathrm{d}s) = H(\mathrm{d}s)\Phi(\mathrm{d}x)$ so that the RHS above equals

$$\begin{aligned}&\mathbb{E}_\Phi\left[\int_{\mathbb{X}}\int_{\mathbb{S}^{n-1}} g(x,s,\Psi-\delta_{x,s}) \prod_{i:x_i\in\Phi_x-\delta_x} H(\mathrm{d}m_i)\Psi(\mathrm{d}x,\mathrm{d}s)\right]\\ &\quad= \mathbb{E}_\Phi\left[\int_{\mathbb{S}^{n-1}}\int_{\mathbb{X}} g(x,s,\Psi-\delta_{x,s}) \prod_{i:x_i\in\Phi_x-\delta_x} H(\mathrm{d}m_i)\Psi(\mathrm{d}x,\mathrm{d}s)\right]\\ &\quad= \mathbb{E}_\Psi\left[\int_{\mathbb{X}} g(x,s,\Psi-\delta_{x,s})\Psi(\mathrm{d}x,\mathrm{d}s)\right]\end{aligned}$$

which proves (3.24) and the results follows.

The proof for the n-th Palm distribution is obtained following the same lines constructing $\Psi'_{\mathbf{x},\mathbf{s}}$ from $\Phi^!_{\mathbf{x}}$ in an analogous way. □

Lemma 3.3. *Let $\tilde{\mu}(A) = \int_{A\times\mathbb{R}_+} s\Psi(\mathrm{d}x\mathrm{d}s)$ where $\Psi = \sum_{j\geq 1}\delta_{(X_j,S_j)}$ is a marked point process obtained by marking $\Phi = \sum_j \delta_{X_j}$ with i.i.d. marks S_j from a distribution H on $\mathbb{R}_+$, then for any measurable $f \geq 0$:*

$$\mathbb{E}\left[e^{-\int_{\mathbb{X}} f(x)\tilde{\mu}(\mathrm{d}x)}\right] = \mathbb{E}\left[\exp\left(\int_{\mathbb{X}} \log\psi(f(x))\Phi(\mathrm{d}x)\right)\right]$$

where $\psi(f(x)) := \int_{\mathbb{R}_+} e^{-sf(x)}H(\mathrm{d}s)$ is the Laplace transform of H evaluated at $f(x)$.

Proof. By the tower property of the expected value, conditioning of Φ and taking the expectation we obtain

$$\begin{aligned}
\mathbb{E}\left[\exp\left\{-\int_{\mathbb{X}} f(x)\widetilde{\mu}(\mathrm{d}x)\right\}\right] &= \mathbb{E}\left[\exp\left\{-\sum_{j\geq 1} S_j f(X_j)\right\}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\exp\left\{-\sum_{j\geq 1} S_j f(X_j))\right\}\Big|\Phi\right]\right] \\
&= \mathbb{E}\left[\prod_{j\geq 1}\mathbb{E}\left[\exp\left\{-S_j f(X_j)\right\}\Big|\Phi\right]\right] \\
&= \mathbb{E}\left[\prod_{j\geq 1}\int_{\mathbb{R}_+} e^{-sf(X_j)}H(\mathrm{d}s)\right] \\
&= \mathbb{E}\left[\exp\left\{\sum_{j\geq 1}\log\left(\int_{\mathbb{R}_+} e^{-sf(X_j)}H(\mathrm{d}s)\right)\right\}\right] \\
&= \mathbb{E}\left[\exp\left(\int_{\mathbb{X}}\log\psi(f(x))\Phi(\mathrm{d}x)\right)\right]
\end{aligned}$$

□

3.C Proofs

3.C.1 Proof of Proposition 3.1

Proof.

$$\begin{aligned}
\mathbb{E}[\widetilde{\mu}(A)] &= \mathbb{E}\left[\int_{A\times\mathbb{R}_+} s\Psi(\mathrm{d}x,\mathrm{d}s)\right] \\
&= \int_{A\times\mathbb{R}_+}\int_{\mathbb{M}(A\times\mathbb{R}_+)} s\mathbf{P}_{\Psi}^{x,s}(\mathrm{d}\nu)M_{\Psi}(\mathrm{d}x,\mathrm{d}s) \\
&= \int_{A\times\mathbb{R}_+} sH(\mathrm{d}s)M_{\phi}(\mathrm{d}x)
\end{aligned}$$

where the second equality follows from Theorem 3.5 with $g(x,s,\Psi) = s$ while the third follows from Proposition 3.1.

By the identity $x^{-1} = \int_{\mathbb{R}_+} e^{-ux}\mathrm{d}u$ we have

$$\mathbb{E}[\tilde{p}(A)] = \mathbb{E}\left[\frac{\widetilde{\mu}(A)}{\widetilde{\mu}(\mathbb{X})}\right] =$$

$$\mathbb{E}\left[\int_{\mathbb{R}_+}\mathrm{d}u\int_{A\times\mathbb{R}_+}\exp\left(-\int_{\mathbb{X}\times\mathbb{R}_+} ut\Psi(\mathrm{d}z,\mathrm{d}t)\right)s\Psi(\mathrm{d}x,\mathrm{d}s)\right].$$

We can further exchange the outermost expectation with the integral with respect to du by Fubini theorem and apply Theorem 3.5 with $g(x, s, \Psi) = \exp\left(-\int_{\mathbb{X}\times\mathbb{R}_+} ut\Psi(\mathrm{d}z, \mathrm{d}t)\right) s$, leading to

$$\mathbb{E}[\tilde{p}(A)] = \int_{\mathbb{R}_+}\int_{A\times\mathbb{R}_+} e^{-us} sH(\mathrm{d}s)\mathbb{E}\left[e^{-\int_{\mathbb{X}\times\mathbb{R}_+} uv\Psi^!_{x,s}(\mathrm{d}z\,\mathrm{d}v)}\right] \mathrm{d}uM_\Phi(\mathrm{d}x)$$

where $\Psi^!_{x,s}$ is the reduced Palm kernel of Ψ.

$$\mathrm{Cov}(\widetilde{\mu}(A), \widetilde{\mu}(B)) = \mathbb{E}[\widetilde{\mu}(A)\widetilde{\mu}(B)] - \mathbb{E}[\widetilde{\mu}(A)]\mathbb{E}[\widetilde{\mu}(B)].$$

Focusing on the first term we get

$$\mathbb{E}[\widetilde{\mu}(A)\widetilde{\mu}(B)] = \mathbb{E}\left[\int_{A\times\mathbb{R}_+}\int_{B\times\mathbb{R}_+} st\Psi(\mathrm{d}x, \mathrm{d}s)\Psi(\mathrm{d}z, \mathrm{d}t)\right]$$

Let $\boldsymbol{s} = (s, t)$ and $\boldsymbol{x} = (x, s)$

$$\mathbb{E}[\widetilde{\mu}(A)\widetilde{\mu}(B)] = \mathbb{E}\left[\int_{(\mathbb{X}\times\mathbb{R}_+)^2} \mathbb{1}_{A\times B}(\boldsymbol{x})st\Psi(\mathrm{d}\boldsymbol{x}, \mathrm{d}\boldsymbol{s})\right]$$

an application of Theorem 3.6 with $g(\boldsymbol{x}, \boldsymbol{s}, \Psi) = \mathbb{1}_{A\times B}(\boldsymbol{x})st$ yields the proof.

□

3.C.2 Proof of Theorem 3.1

Proof. The focus of our study is the characteristic functional of $\widetilde{\mu}$ given $\boldsymbol{Y} = \boldsymbol{y}$ and $U_n = u$:

$$\mathbb{E}\left[e^{-\int_{\mathbb{X}} f(z)\widetilde{\mu}(\mathrm{d}z)} \,\middle|\, \boldsymbol{Y} = \boldsymbol{y}, U_n = u\right] = \frac{\mathbb{E}\left[e^{-\int_{\mathbb{X}} f(z)\mu(\mathrm{d}z)}\mathsf{P}(\boldsymbol{Y} \in \mathrm{d}\boldsymbol{y}, U \in \mathrm{d}u \mid \widetilde{\mu})\right]}{\mathbb{E}\left[\mathsf{P}(\boldsymbol{Y} \in \mathrm{d}\boldsymbol{y}, U \in \mathrm{d}u \mid \widetilde{\mu})\right]}$$

First, observe that the expression in the denominator is obtained as a special case of the expression of the numerator by letting $f(x) = 0$. Focusing on the numerator

$$\begin{aligned}
&\mathbb{E}\left[e^{-\int_{\mathbb{X}} f(z)\widetilde{\mu}(\mathrm{d}z)}\mathsf{P}(\boldsymbol{Y} \in \mathrm{d}\boldsymbol{y}, U \in \mathrm{d}u \mid \widetilde{\mu})\right] \\
&= \mathbb{E}\left[e^{-\int_{\mathbb{X}} f(z)\widetilde{\mu}(\mathrm{d}z)}\frac{u^{n-1}}{\Gamma(n)}\prod_{j=1}^k \widetilde{\mu}(\mathrm{d}y_j^*)^{n_j} e^{-Tu}\right] \\
&= \frac{u^{n-1}}{\Gamma(n)}\mathbb{E}\left[e^{-\int_{\mathbb{X}}(f(z)+u)\widetilde{\mu}(\mathrm{d}z)}\prod_{j=1}^k \widetilde{\mu}(\mathrm{d}y_j^*)^{n_j}\right] \\
&= \frac{u^{n-1}}{\Gamma(n)}\mathbb{E}\left[\int_{(\mathbb{X}\times\mathbb{R}_+)^k} e^{-\int_{\mathbb{X}}(f(z)+u)\widetilde{\mu}(\mathrm{d}z)}\prod_{j=1}^k s_j^{n_j}\delta_{y_j^*}(x_j)\Psi(\mathrm{d}x_j, \mathrm{d}s_j)\right]
\end{aligned}$$

We are now in place to apply Theorem 3.6 on the marked point process Ψ. Defining

$$g(\boldsymbol{x}, \boldsymbol{s}, \Psi) = e^{-\int_{\mathbb{X}}(f(z)+u)\widetilde{\mu}(\mathrm{d}z)}\prod_{j=1}^k s_j^{n_j}\delta_{y_j^*}(x_j),$$

we obtain that

$$\begin{aligned}
&\mathbb{E}\left[e^{-\int_{\mathbb{X}} f(z)\widetilde{\mu}(\mathrm{d}z)}\mathsf{P}(\boldsymbol{Y} \in \mathrm{d}\boldsymbol{y}, U \in \mathrm{d}u \mid \widetilde{\mu})\right] = \\
&\qquad \frac{u^{n-1}}{\Gamma(n)}\int_{(\mathbb{X}\times\mathbb{R}_+)^k} \mathbb{E}_{\Psi\sim\mathbf{P}_\Psi^{\boldsymbol{x},\boldsymbol{s}}}\left[g(\boldsymbol{x}, \boldsymbol{s}, \Psi)\right] M_{\Phi^k}(\mathrm{d}\boldsymbol{x})H^k(\mathrm{d}\boldsymbol{s})
\end{aligned}$$

where we have used $M_{\Psi^k}(\mathrm{d}\boldsymbol{x}, \mathrm{d}\boldsymbol{s}) = M_{\Phi^k}(\mathrm{d}\boldsymbol{x})H^k(\mathrm{d}\boldsymbol{s})$ as in Proposition 3.1.

By the properties of the Palm distribution, $\Psi \sim \mathbf{P}_{\Psi}^{\mathbf{x},\mathbf{s}}$ has the same law of $\sum_{j=1}^{k} \delta_{x_j,s_j} + \Psi^!_{\mathbf{x},\mathbf{s}}$. Moreover, following Proposition 3.2, $\Psi^!_{\mathbf{x},\mathbf{s}} = \sum_j \delta_{\tilde{X}_j,\tilde{S}_j}$ where $\Phi^!_{\mathbf{x}} := \sum \delta_{\tilde{X}} \sim \mathbf{P}_{\Phi^!}^{\mathbf{x}}$ and the $\tilde{S}_j$'s are iid from H. Hence

$$\mathbb{E}_{\Psi \sim \mathbf{P}_{\Psi}^{\mathbf{x},\mathbf{s}}}\left[g(\boldsymbol{x}, \boldsymbol{s}, \Psi)\right] = \mathbb{E}\left[g\left(\boldsymbol{x}, \boldsymbol{s}, \sum_{j=1}^{k} \delta_{(x_j,s_j)} + \Psi^!_{\mathbf{x},\mathbf{s}}\right)\right]$$

where the expected value on the right hand side is taken with respect to $\Psi^!_{\mathbf{x},\mathbf{s}}$. Let $\widetilde{\mu}^!_{\boldsymbol{y}^*}(A) := \int_{A \times \mathbb{R}_+} s \Psi^!_{\mathbf{x},\mathbf{s}}(\mathrm{d}x, \mathrm{d}s)$, then we have

$$g\left(\boldsymbol{x}, \boldsymbol{s}, \sum_{j=1}^{k} \delta_{x_j,s_j} + \Psi^!_{\mathbf{x},\mathbf{s}}\right) = \\ \exp\left(-\int_{\mathbb{X}} (f(z)+u)\left(\sum_{j=1}^{k} s_j \delta_{x_j}(\mathrm{d}z) + \widetilde{\mu}^!(\mathrm{d}z)\right)\right) \prod_{j=1}^{k} s_j^{n_j} \delta_{y_j^*}(x_j)$$

Hence,

$$\begin{aligned}
&\mathbb{E}\left[e^{-\int_{\mathbb{X}} f(z) \widetilde{\mu}(\mathrm{d}z)} \mathsf{P}(\boldsymbol{Y} \in \mathrm{d}\boldsymbol{y}, U \in \mathrm{d}u \,|\, \widetilde{\mu})\right] = \\
&= \frac{u^{n-1}}{\Gamma(n)} \int_{(\mathbb{X} \times \mathbb{R}_+)^k} \mathbb{E}\left[\exp\left(-\int_{\mathbb{X}} (f(z)+u)\left(\sum_{j=1}^{k} s_j \delta_{x_j}(\mathrm{d}z) + \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)\right)\right) \prod_{j=1}^{k} s_j^{n_j} \delta_{y_j^*}(x_j)\right] \\
&\qquad \times M_{\Phi^k}(\mathrm{d}\boldsymbol{x}) H^k(\mathrm{d}\boldsymbol{s}) \\
&= \frac{u^{n-1}}{\Gamma(n)} \int_{(\mathbb{X} \times \mathbb{R}_+)^k} \mathbb{E}\left[e^{-\int_{\mathbb{X}} (f(z)+u) \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right] \exp\left(-\int_{\mathbb{X}} (f(z)+u)\left(\sum_{j=1}^{k} s_j \delta_{x_j}(\mathrm{d}z)\right)\right) \\
&\qquad \times \prod_{j=1}^{k} s_j^{n_j} \delta_{y_j^*}(x_j) M_{\Phi^k}(\mathrm{d}\boldsymbol{x}) H^k(\mathrm{d}\boldsymbol{s}) \\
&= \frac{u^{n-1}}{\Gamma(n)} \mathbb{E}\left[e^{-\int_{\mathbb{X}} (f(z)+u) \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right] M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*) \prod_{j=1}^{k} \int_{\mathbb{R}_+} e^{-(f(y_k^*)+u)s_j} s_j^{n_j} H(\mathrm{d}s_j)
\end{aligned}$$

Setting $f = 0$ yields the denominator

$$\mathbb{E}\left[\mathsf{P}(\boldsymbol{Y} \in \mathrm{d}\boldsymbol{y}, U \in \mathrm{d}u \,|\, \widetilde{\mu})\right] = \\ \frac{u^{n-1}}{\Gamma(n)} \mathbb{E}\left[e^{-\int_{\mathbb{X}} u \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right] M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*) \prod_{j=1}^{k} \int_{\mathbb{R}_+} e^{-u s_j} s_j^{n_j} H(\mathrm{d}s_j) \tag{3.25}$$

Defining $\kappa(u, n_j) = \int_{\mathbb{R}_+} e^{-u s_j} s_j^{n_j} H(\mathrm{d}s_j)$, we obtain

$$\mathbb{E}\left[\exp \int_{\mathbb{X}} -f(z) \widetilde{\mu}(\mathrm{d}z) \,\middle|\, \boldsymbol{Y} = \boldsymbol{y}, U_n = u\right] = \\ \frac{\mathbb{E}\left[e^{-\int_{\mathbb{X}} (f(z)+u) \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right]}{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right]} \prod_{j=1}^{k} \frac{e^{-f(y_j^*)s_j} e^{-u s_j} s_j^{n_j}}{\kappa(u, n_j)} H(\mathrm{d}s_j). \tag{3.26}$$

The first term corresponds to the Laplace transform of $\widetilde{\mu}$, while in the second term we recongize the Laplace transform of $\sum_{j=1}^{k} S_j^* \delta_{Y_j^*}$ where $S_j^* \sim f_{S_j^*}(s) \propto e^{-us_j} s_j^{n_j} H(\mathrm{d}s_j)$.

The conditional distribution of u follows easily from (3.25) conditioning on $\boldsymbol{y}$. □

3.C.3 Proof of Theorem 3.2

The proof follows by integrating (3.25) with respect to u.

3.C.4 Proof of Theorem 3.3

Proof. In order to prove the theorem consider a sufficiently small $\epsilon > 0$, so that the balls $B_\epsilon(y_1^*), \ldots, B_\epsilon(y_k^*)$ are all disjoint and observe that

$$\mathsf{P}(Y_{n+1} \in A \,|\, \boldsymbol{Y} = \boldsymbol{y}) = \lim_{\epsilon \to 0} \frac{\mathsf{P}(y_{n+1} \in A, \boldsymbol{Y} \in \times_{j=1}^{k} B_\epsilon^{n_j}(y_j^*))}{\mathsf{P}(\boldsymbol{Y} \in \times_{j=1}^{k} B_\epsilon^{n_j}(y_j^*))}. \tag{3.27}$$

Now set $A^* := A \setminus \cup_{j=1}^{k} B_\epsilon(y_j^*)$, we then obtain that the ratio in the previous limit equals

$$\begin{aligned}\frac{\mathsf{P}(Y_{n+1} \in A, \boldsymbol{Y} \in \times_{j=1}^{k} B_\epsilon^{n_j}(y_j^*))}{\mathsf{P}(\boldsymbol{Y} \in \times_{j=1}^{k} B_\epsilon^{n_j}(y_j^*))} &= \frac{\mathsf{P}(Y_{n+1} \in A^*, \boldsymbol{Y} \in \times_{j=1}^{k} B_\epsilon^{n_j}(y_j^*))}{\mathsf{P}(\boldsymbol{Y} \in \times_{j=1}^{k} B_\epsilon^{n_j}(y_j^*))} \\ &\quad + \sum_{j=1}^{k} \frac{\mathsf{P}(Y_{n+1} \in B_\epsilon(y_j^*) \cap A, \boldsymbol{Y} \in \times_{j=1}^{k} B_\epsilon^{n_j}(y_j^*))}{\mathsf{P}(\boldsymbol{Y} \in \times_{j=1}^{k} B_\epsilon^{n_j}(y_j^*))}.\end{aligned}$$

Now we exploit Theorem 3.2 to evaluate the previous expressions, in particular for the first one we get

$$\begin{aligned}&\frac{\mathsf{P}(Y_{n+1} \in A^*, \boldsymbol{Y} \in \times_{j=1}^{k} B_\epsilon^{n_j}(y_j^*))}{P(\boldsymbol{Y} \in \times_{j=1}^{k} B_\epsilon^{n_j}(y_j^*))} \\ &\quad = \int_{A^*} \int_{\mathbb{R}_+} \frac{u}{n} \frac{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u \widetilde{\mu}^!_{(\boldsymbol{y}^*, y)}(\mathrm{d}z)}\right]}{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right]} \frac{m_{\Phi^{k+1}}(\mathrm{d}\boldsymbol{y}^*, y)}{m_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*)} \kappa(u,1) f_{U_n}(u|\boldsymbol{y}) \mathrm{d}u P_0(\mathrm{d}y) \\ &\qquad + o\Big(\prod_{j=1}^{k} P_0(B_\epsilon(y_j^*))\Big)\end{aligned}$$

analogously, for any $j = 1, \ldots, k$, one has

$$\begin{aligned}&\frac{\mathsf{P}(Y_{n+1} \in B_\epsilon(y_j^*) \cap A, \boldsymbol{Y} \in \times_{j=1}^{k} B_\epsilon^{n_j}(y_j^*))}{\mathsf{P}(\boldsymbol{Y} \in \times_{j=1}^{k} B_\epsilon^{n_j}(y_j^*))} \\ &\quad = \frac{1}{P_0(B_\epsilon(y_j^*))} \int_{A^* \cap B_\epsilon(y_j^*)} \int_{\mathbb{R}_+} \frac{u}{n} \frac{\kappa(u, n_j+1)}{\kappa(u, n_j)} f_{U_n}(u|\boldsymbol{y}) \mathrm{d}u P_0(\mathrm{d}y) \\ &\qquad + o\Big(\prod_{j=1}^{k} P_0(B_\epsilon(y_j^*))\Big).\end{aligned}$$

By letting $\epsilon \to 0$, we obtain the following result:

$$\begin{aligned}\mathsf{P}(Y_{n+1} \in A \,|\, \boldsymbol{Y} = \boldsymbol{y}) &= \int_{\mathbb{R}_+} \frac{u}{n} \sum_{j=1}^{k} \frac{\kappa(u, n_j+1)}{\kappa(u, n_j)} \delta_{y_j^*}(A) f_{U_n}(u \,|\, \boldsymbol{y}) \mathrm{d}u + \\ &\int_A \int_{\mathbb{R}_+} \frac{u}{n} \kappa(u,1) \frac{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u \mu^!_{(\boldsymbol{y}^*, y)}(\mathrm{d}z)}\right]}{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u \mu^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right]} \frac{M_{\Phi^{k+1}}(\mathrm{d}\boldsymbol{y}^*, y)}{M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*)} f_{U_n}(u \,|\, \boldsymbol{y}) \mathrm{d}u \mathrm{d}y.\end{aligned} \tag{3.28}$$

where we used the Lebesgue Theorem and the fact that P_0 is non-atomic. We can now exploit (Kallenberg, 2021, Lemma 8.16) to augment the probability space with the inclusion of a random variable U_n, such that the joint distribution of Y_{n+1} and U_n, conditionally given $\boldsymbol{Y} = \boldsymbol{y}$, is

$$\begin{aligned}\mathsf{P}(Y_{n+1} \in A, U_n \in B | \boldsymbol{Y} = \boldsymbol{y}) &= \int_B \frac{u}{n} \sum_{j=1}^{k} \frac{\kappa(u, n_j + 1)}{\kappa(u, n_j)} \delta_{y_j^*}(A) f_{U_n}(u \,|\, \boldsymbol{y}) \mathrm{d}u \\ &+ \int_A \int_B \frac{u}{n} \kappa(u, 1) \frac{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u \mu^!_{(\boldsymbol{y}^*, y)}(\mathrm{d}z)}\right]}{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u \mu^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right]} \frac{M_{\Phi^{k+1}}(\mathrm{d}\boldsymbol{y}^*, y)}{M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*)} f_{U_n}(u \,|\, \boldsymbol{y}) \mathrm{d}y \mathrm{d}u,\end{aligned} \tag{3.29}$$

now the result follows thanks to the Bayes Theorem. □

3.C.5 Proof of Proposition 3.2

Proof. First, observe that $\widetilde{\mu}^!_{\boldsymbol{y}^*}(A) = \sum_{j \geq 1} S_j \delta_{X_j}(A) = \int_{A \times \mathbb{R}_+} s \Psi^!_{\mathbf{y}^*}$, where $\Psi^!_{\mathbf{y}^*}$ is obtained by marking $\Phi^!_{\mathbf{y}^*}$ (the reduced Palm version of Φ) with i.i.d. marks from H. So that, denoting with $n^!$ the number of points in $\Phi^!_{\mathbf{y}^*}$ we have

$$\begin{aligned}\mathbb{E}\left[e^{-\int_X u \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}x)}\right] &= \mathbb{E}\left[\mathbb{E}\left[\exp\left(-\sum_{j \geq 1} U_n S_j \delta_{X_j}(\mathbb{X})\right) \mid \Phi^!_{\mathbf{y}^*}\right]\right] \\ &= \mathbb{E}\left[\prod_{j=1}^{n^!} \int_{\mathbb{R}_+} e^{-U_n s} H(\mathrm{d}s)\right] \\ &= \mathbb{E}[\psi(U_n)^{n^!}]\end{aligned}$$

Let $q_r = P(n^! = r)$, $r = 0, 1, \ldots$ the probability mass faction of the number of points in $\Phi^!_{\mathbf{y}^*}$, so that $\mathbb{E}[\psi(u)^{n^!}] = \sum_{r \geq 0} \psi(u)^r q_r$, then we can write the marginal as

$$\begin{aligned}\mathsf{P}(\boldsymbol{Y}^* \in \mathrm{d}y^*, \boldsymbol{N} = \boldsymbol{n}) &= \int_{\mathbb{R}_+} \frac{u^{n-1}}{\Gamma(n)} \mathbb{E}\left[e^{-\int_{\mathbb{X}} u \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right] \prod_{j=1}^{k} \kappa(u, n_j) \mathrm{d}u \, M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*) \\ &= \int_{\mathbb{R}_+} \frac{u^{n-1}}{\Gamma(n)} \sum_{r \geq 0} \psi(u)^r q_r \prod_{j=1}^{k} \kappa(u, n_j) \mathrm{d}u \, M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*) \\ &= \sum_{r \geq 0} q_r \int_{\mathbb{R}_+} \frac{u^{n-1}}{\Gamma(n)} \psi(u)^r \prod_{j=1}^{k} \kappa(u, n_j) \mathrm{d}u \, M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*)\end{aligned}$$

where the third equality follows from Fubini's theorem.

Then, by the definition of $V(n_1, \ldots, n_k; r)$ we have

$$\mathsf{P}(K_n = k, \boldsymbol{Y}^* \in \mathrm{d}\boldsymbol{y}^*) = \frac{1}{k!} \sum_{r=0}^{\infty} \left(\sum_{n_1 + \cdots + n_k = n} \binom{n}{n_1 \cdots n_k} V(n_1, \ldots, n_k; r) \right) q_r \; M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*)$$

□

3.C.6 Proof of Corollary 3.1

We have that $\psi(u) = (u+1)^{-\alpha}$ and $\kappa(u, n) = (\alpha)_n(u+1)^{-(n+\alpha)}$, where $(\alpha)_n := \Gamma(\alpha + n)/\Gamma(\alpha)$ denotes the rising factorial or Pochhammer symbol. Moreover,

$$\begin{aligned} V(n_1, \dots, n_k; r) &= \int_{\mathbb{R}_+} \frac{u^{n-1}}{\Gamma(n)} \psi(u)^r \prod_{j=1}^{k} \kappa(u, n_j) \mathrm{d}u \\ &= \frac{1}{\Gamma(n)} \prod_{j=1}^{k} (\alpha)_{n_j} \int \frac{u^{n-1}}{(u+1)^{n+\alpha k+\alpha r}} \mathrm{d}u \\ &= \frac{1}{\Gamma(n)} \prod_{j=1}^{k} (\alpha)_{n_j} \frac{\Gamma\left((k+r)\alpha\right)}{\Gamma\left((k+r)\alpha+n\right)} \end{aligned}$$

Hence

$$\mathsf{P}(\boldsymbol{Y}^* \in \mathrm{d}y^*, \boldsymbol{N} = \boldsymbol{n}) = \frac{1}{\Gamma(n)} \prod_{j=1}^{k} (\alpha)_{n_j} \left(\sum_{r \geq 0} q_r \frac{\Gamma\,(k+r)\alpha)}{\Gamma\left((k+r)\alpha+n\right)} \right) M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*),$$

and

$$\begin{aligned} \mathsf{P}(K_n = k, \boldsymbol{Y}^* \in \mathrm{d}\boldsymbol{y}^*) &= \frac{1}{\Gamma(n)} \left(\sum_{r \geq 0} q_r \frac{\Gamma\,(k+r)\alpha)}{\Gamma\left((k+r)\alpha+n\right)} \right) M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*) \\ &\quad \times \frac{1}{k!} \sum_{n1+\cdots+n_k=n} \binom{n}{n_1 \cdots n_k} \prod_{j=1}^{k} (\alpha)_{n_j} \\ &= \frac{1}{\Gamma(n)} \alpha^k S_{n,k}^{-1,k} \left(\sum_{r \geq 0} q_r \frac{\Gamma\,(k+r)\alpha)}{\Gamma\left((k+r)\alpha+n\right)} \right) M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*) \end{aligned}$$

where $S_{n,k}^{-1,k}$ denotes the generalized Stirling number.

3.D Details about the Poisson point process Examples

Theorem 3.7. *Let Φ be a Poisson point process with intensity $\nu(x)\mathrm{d}x$. Then the random measure $\widetilde{\mu}$ equals the distribution of*

$$\sum_{j \geq 1} S'_j \delta_{X'_j}$$

where $\Psi' := \sum_{j \geq 1} \delta_{(X'_j, S'_j)}$ is a marked point process whose unmarked point process $\Phi' := \sum_{j \geq 1} \delta_{X'_j}$ is a Poisson process with intensity given by $\Psi(u)\lambda(\mathrm{d}x)$ and the marks S'_j are i.i.d. with distribution

$$H'(\mathrm{d}s) := \frac{e^{-su} H(\mathrm{d}s)}{\int_{\mathbb{R}^+} e^{-su} H(\mathrm{d}s)}.$$

Proof. The random measure $\widetilde{\mu}'$ in Theorem 3.1 has Laplace functional (3.9). By Lemma 3.3, we have that the numerator can be expressed as

$$\mathbb{E}\left[\exp\left\{ -\int_{\mathbb{X}} (f(z) + u) \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z) \right\} \right] = \mathbb{E}\left[\exp\left(\int_{\mathbb{X}} \int_{\mathbb{R}_+} e^{-s(f(z)+u)} H(\mathrm{d}s) \Phi^!_{\mathbf{y}^*}(\mathrm{d}x) \right) \right]$$

Where $\Phi^!_{\mathbf{y}^*}$ is the reduced Palm version of Φ at $\boldsymbol{y}^*$. Thanks to the properties of the Poisson process, we have that $\Phi^!_{\mathbf{y}^*}$ equals to Φ in distribution. Hence, the expectation can be evaluated using the Lévy-Khintchine representation:

$$\mathbb{E}\left[\exp\left\{-\int_{\mathbb{X}}(f(z)+u)\widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)\right\}\right] = \exp\left(-\int_{\mathbb{X}}\int_{\mathbb{R}_+} 1 - e^{-s(f(z)+u)}H(\mathrm{d}s)\nu(x)\mathrm{d}x\right).$$

The same expression can be derived for the denominator setting $f = 0$. Combining numerator and denominator we have that

$$\begin{aligned}\mathbb{E}&\left[\exp\int_{\mathbb{X}} -f(z)\widetilde{\mu}'(\mathrm{d}z)\right]\\ &= \exp\left(-\int_{\mathbb{X}}\int_{\mathbb{R}_+} -e^{-s(f(z)+u)} + e^{-su}H(\mathrm{d}s)\nu(x)\mathrm{d}x\right)\\ &= \exp\left(-\int_{\mathbb{X}}\int_{\mathbb{R}_+}\left(1-e^{-sf(z)}\right)e^{-su}H(\mathrm{d}s)\nu(x)\mathrm{d}x\right)\end{aligned}$$

multiplying and dividing by $\psi(u) := \int_{\mathbb{R}_+} e^{-su}H(\mathrm{d}s)$ we can recognize the Laplace transform of the random measure

$$\widetilde{\mu}' = \sum_{j\geq 1} S'_j\delta_{X'_j}$$

where the S'_j's are i.i.d. random variables with density $\psi(u)^{-1}1e^{-su}H(\mathrm{d}s)$ and $\sum_j \delta_{X'_j}$ is a Poisson process with intensity $\psi(u)\nu(x)\mathrm{d}x$. □

3.E Details about the Gibbs point process Examples

Proposition 3.4. *The k-th moment measure of a Gibbs point process with density f_Φ with respect to a Poisson point process with intensity λ is*

$$M_{\Phi^k}(B) = \ E_N\left[f_\Phi\left(N + \sum_{j=1}^k \delta_{x_j}\right)\right]$$

Proof. Let $B = B_1 \times \cdots \times B_k$, $B_j \in \mathcal{X}$, then

$$\begin{aligned}M_{\Phi^k}(B) &= \int_{\mathcal{M}(\mathbb{X})}\prod_{j=1}^k \nu(B_j)\mathcal{P}_\Phi(\mathrm{d}\nu) = \int_{\mathcal{M}(\mathbb{X})}\prod_{j=1}^k \nu(B_j)f_\Phi(\nu)\mathcal{P}_N(\mathrm{d}\nu)\\ &= \mathbb{E}_N\left[\int_{\mathbb{X}^k}\prod_{j=1}^k I[x_j\in B_j]f_\Phi(N)N^k(\mathrm{d}x_1,\dots,\mathrm{d}x_k)\right]\\ &= \int_{\mathbb{X}^k}\mathbb{E}_N\left[f_\Phi\left(N+\sum_{j=1}^k\delta_{x_j}\right)\right]\lambda(\mathrm{d}x_1)\cdots\lambda(\mathrm{d}x_k)\end{aligned}$$

Where the last equation follows from applying the CLM formula to the Poisson process N, for which $N^!_{\mathbf{x}} \sim N$ and the fact that $M^k_N = \lambda^k$. □

Proposition 3.5. *The k-th reduced Palm distribution of a Gibbs point process with density f_Φ with respect to a Poisson point process with intensity λ is the distribution of another Gibbs point process $\Phi^!_{\mathbf{x}}$ with density*

$$f_{\Phi^!_{\mathbf{x}}}(\nu) = \frac{f_\Phi(\nu + \sum_{j=1}^k \delta_{x_j})}{\mathbb{E}_N\left[f_\Phi(N + \sum_{j=1}^k\delta_{x_j})\right]}$$

with respect to N.

Theorem 3.8. *If Φ is a Gibbs point process with density f_Φ, the random measure $\widetilde{\mu}'$ equals the distribution of*

$$\sum_{j\geq 1} S'_j \delta_{X'_j}$$

where $\Psi' := \sum_{j\geq 1}\delta_{(X'_j,S'_j)}$ is a marked point process whose unmarked point process $\Phi' := \sum_{j\geq 1}\delta_{X'_j}$ is of Gibbs type with density with respect to the Poisson process N given by

$$f_{\Phi'}(\nu) := \frac{\exp\left\{\int_{\mathbb{X}} \log\Big(\int_{\mathbb{R}_+} e^{-su}H(\mathrm{d}s)\Big)\nu(\mathrm{d}x)\right\} f_\Phi(\nu + \sum_{j=1}^k \delta_{y_j^*})}{\mathbb{E}_N\left[\exp\left\{\int_{\mathbb{X}} \log\Big(\int_{\mathbb{R}_+} e^{-su}H(\mathrm{d}s)\Big)N(\mathrm{d}x)\right\} f_\Phi(N + \sum_{j=1}^k \delta_{y_j^*})\right]},$$

and the marks S'_j are i.i.d. with distribution

$$H'(\mathrm{d}s) := \frac{e^{-su}H(\mathrm{d}s)}{\int_{\mathbb{R}^+} e^{-su}H(\mathrm{d}s)}.$$

Proof. The random measure $\tilde{\mu}'$ in Theorem 3.1 has Laplace functional (3.9), in order to characterize its distribution we first evaluate the numerator in (3.9). From Lemma 3.3, we have

$$\mathbb{E}\left[\exp\left\{-\int_{\mathbb{X}}(f(z)+u)\widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)\right\}\right] = \mathbb{E}\left[\prod_{j\geq 1}\int_{\mathbb{R}_+} e^{-s(f(X_j)+u)}H(\mathrm{d}s)\right]$$

where the X_j's are the support points of $\widetilde{\mu}^!_{\boldsymbol{y}^*}$. We now exploit Proposition 3.5 to evaluate the last expected value. Indeed, by virtue of this proposition, $\Phi^!_{\mathbf{y}^*}$ is again a Gibbs point process with density with respect to the Poisson process N given by

$$f_{\Phi^!_{\mathbf{y}^*}}(\nu) = \frac{f_\Phi(\nu + \sum_{j=1}^k \delta_{y_j^*})}{\mathbb{E}_N\left[f_\Phi(N + \sum_{j=1}^k \delta_{y_j^*})\right]}, \quad \nu \in \mathbb{M}(\mathbb{X}).$$

As a consequence one has:

$$\begin{aligned}\mathbb{E}\left[\exp\left\{-\int_{\mathbb{X}}(f(z)+u)\widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}z)\right\}\right] &= \mathbb{E}\left[\exp\left\{\sum_{j\geq 1}\log\Big(\int_{\mathbb{R}_+} e^{-s(f(X_j)+u)}H(\mathrm{d}s)\Big)\right\}\right]\\ &= \mathbb{E}\left[\exp\left\{\int_{\mathbb{X}}\log\Big(\int_{\mathbb{R}_+} e^{-s(f(x)+u)}H(\mathrm{d}s)\Big)\Phi^!_{\mathbf{y}^*}(\mathrm{d}x)\right\}\right]\\ &= \frac{\mathbb{E}_N\left[\exp\left\{\int_{\mathbb{X}}\log\Big(\int_{\mathbb{R}_+} e^{-s(f(x)+u)}H(\mathrm{d}s)\Big)N(\mathrm{d}x)\right\} f_\Phi(N + \sum_{j=1}^k \delta_{y_j^*})\right]}{\mathbb{E}_N\left[f_\Phi(N + \sum_{j=1}^k \delta_{y_j^*})\right]}.\end{aligned}$$

The previous expression corresponds to the numerator in (3.9), while the denominator follows by considering the function $f = 0$, thus one has

$$\begin{aligned}&\mathbb{E}\left[\exp\int_{\mathbb{X}} -f(z)\tilde{\mu}'(\mathrm{d}z)\right]\\ &\quad= \frac{\mathbb{E}_N\left[\exp\left\{\int_{\mathbb{X}}\log\Big(\int_{\mathbb{R}_+} e^{-s(f(x)+u)}H(\mathrm{d}s)\Big)N(\mathrm{d}x)\right\} f_\Phi(N + \sum_{j=1}^k \delta_{y_j^*})\right]}{\mathbb{E}_N\left[\exp\left\{\int_{\mathbb{X}}\log\Big(\int_{\mathbb{R}_+} e^{-su}H(\mathrm{d}s)\Big)N(\mathrm{d}x)\right\} f_\Phi(N + \sum_{j=1}^k \delta_{y_j^*})\right]}.\end{aligned} \tag{3.30}$$

The last expression in (3.30) may be rearranged as follows

$$\mathbb{E}_N \left[\exp \left\{ \int_{\mathbb{X}} \log \Big(\int_{\mathbb{R}_+} e^{-sf(x)} H'(\mathrm{d}s) \Big) N(\mathrm{d}x) \right\} f_{\Phi'}(N) \right]$$

where $f_{\Phi'}$ is a density with respect to a Poisson process N defined as

$$f_{\Phi'}(\nu) := \frac{\exp\left\{ \int_{\mathbb{X}} \log \Big(\int_{\mathbb{R}_+} e^{-su} H(\mathrm{d}s) \Big) \nu(\mathrm{d}x) \right\} f_\Phi(\nu + \sum_{j=1}^k \delta_{y_j^*})}{\mathbb{E}_N \left[\exp\left\{ \int_{\mathbb{X}} \log \Big(\int_{\mathbb{R}_+} e^{-su} H(\mathrm{d}s) \Big) N(\mathrm{d}x) \right\} f_\Phi(N + \sum_{j=1}^k \delta_{y_j^*}) \right]}$$

and H' is a new measure on the positive real line $\mathbb{R}_+$ defined as

$$H'(\mathrm{d}s) := \frac{e^{-su} H(\mathrm{d}s)}{\int_{\mathbb{R}^+} e^{-su} H(\mathrm{d}s)}$$

which is an exponential tilting of H. As a consequence we can conclude that $\widetilde{\mu}'$ is a random measure that can be represented as follows

$$\widetilde{\mu}' \overset{d}{=} \int_{A \times \mathbb{R}_+} s \Psi'(\mathrm{d}x, \mathrm{d}s), \;\; \Psi' := \sum_{j \geq 1} \delta_{(X_j', S_j')}$$

and Ψ' is obtained by marking $\Phi^!_{\boldsymbol{y}^*}$ with i.i.d. marks having distribution H'. □

3.E.1 Marginal distribution under a Gibbs point process

Consider now the marginal distribution in Theorem 3.2. By propositions 3.4 and 3.5, we have that

$$\begin{aligned} \mathsf{P}(\boldsymbol{Y} \in \mathrm{d}\boldsymbol{y} \,|\, U_n) &\propto \prod_{j=1}^k \kappa(U_n, n_j) \mathbb{E}\left[e^{-\int_{\mathbb{X}} U_n \widetilde{\mu}^!_{\boldsymbol{y}^*}(\mathrm{d}x)} \right] M_{\Phi^k}(\mathrm{d}\boldsymbol{y}^*) \\ &= \prod_{j=1}^k \kappa(u, n_j) \mathbb{E}\left[\exp\left(\int_{\mathbb{X}} \log \psi(u) N(\mathrm{d}x) \right) f_{\Phi^!_{\mathbf{y}^*}}(N) \right] \mathbb{E}\left[f_{\Phi^!_{\mathbf{y}^*}}(N) \right] \\ &= \prod_{j=1}^k \kappa(u, n_j) \mathbb{E}\left[\exp\left(\int_{\mathbb{X}} \log \psi(u) N(\mathrm{d}x) \right) f_\Phi(N + \sum_{j=1}^k \delta_{y_j^*}) \right] \end{aligned}$$

3.F Details about the DPP Examples

Theorem 3.9. *Assume that Φ is a DPP with kernel K. Moreover assume that its eigenvalues λ_j in (3.5) are all strictly smaller than one. Then, the random measure $\widetilde{\mu}$ equals the distribution of*

$$\sum_{j \geq 1} S_j' \delta_{X_j'}$$

where $\Psi' := \sum_{j \geq 1} \delta_{(X_j', S_j')}$ is a marked point process whose unmarked point process $\Phi' := \sum_{j \geq 1} \delta_{X_j'}$ is a DPP with density with respect to the unit rate Poisson process on S given by $f_\phi(\nu) \propto \det[C'(x_i, x_j)]_{(x_i, x_j) \in \nu}$, where

$$C'(x, y) = \psi(u) \left[C(x, y) - \sum_{i,j=1}^k \left(C^{-1}_{\mathbf{y}^*} \right)_{i,j} C(x, y_i^*) C(y, y_j^*) \right],$$

and the marks S'_j are i.i.d. with distribution

$$H'(\mathrm{d}s) := \frac{e^{-su}H(\mathrm{d}s)}{\int_{\mathbb{R}^+} e^{-su}H(\mathrm{d}s)}.$$

Proof. Since by hypothesis the DPP Φ has density with respect to the Poisson process, it is possible to apply Theorem 3.8 also in this case, so that the point process Φ' has unnormalized density

$$q_{\Phi'}(\nu) = \psi(u)^{n_\nu} f_\Phi\left(\nu + \sum_{j=1}^{k} \delta_{y_j^*}\right)$$

Let $\nu := \sum_j^{n_\nu} \delta_{x_j}$, then the density f_Φ equals to the determinant of the matrix

$$\left[\begin{array}{ccc|ccc}
C(y_1^*, y_1^*) & \cdots & C(y_1^*, y_k^*) & C(y_1^*, x_1) & \cdots & C(y_1^*, x_{n_\nu}) \\
\vdots & & \vdots & \vdots & & \vdots \\
C(y_k^*, y_1^*) & \cdots & C(y_k^*, y_k^*) & C(y_k^*, x_1) & \cdots & C(y_k^*, x_{n_\nu}) \\
\hline
C(x_1, y_1^*) & \cdots & C(x_1, y_k^*) & C(x_1, x_1) & \cdots & C(x_1, x_{n_\nu}) \\
\vdots & & \vdots & \vdots & & \vdots \\
C(x_{n_\nu}, y_1^*) & \cdots & C(x_{n_\nu}, y_k^*) & C(x_{n_\nu}, x_1) & \cdots & C(x_{n_\nu}, x_{n_\nu})
\end{array}\right]$$

Let $C_{\mathbf{y}^*}$ denote the upper left block, C_{xy} the bottom left one and C_{xx} the bottom right one. Then, thanks to Schur's determinant identity, and ignoring the terms that do not depend on ν we have

$$q_{\Phi'}(\nu) = \psi(u)^{n_\nu} \det(C_{xx} - C_{xy} C_{\mathbf{y}^*}^{-1} C_{xy}^T).$$

Let

$$C'(x, y) = \psi(u)\left[C(x, y) - \sum_{i,j=1}^{k} \left(C_{\mathbf{y}^*}^{-1}\right)_{i,j} C(x, y_i^*) C(y, y_j^*)\right],$$

then it is easy to see that $q_{\Phi'}(\nu) = \det[C'(x_i, x_j)]_{(x_i, x_j) \in \nu}$. Therefore, we can conclude that Φ' is a DPP □

3.G Details about the shot-noise Cox process example

3.G.1 Auxiliary Results

Lemma 3.4. *Let Φ be a shot-noise Cox process. Then the reduced Palm distribution of Φ at $\boldsymbol{x} = (x_1, \ldots, x_k)$ equals the law of the point process $\Phi' + \sum_{j=1}^{k} \Phi_{\zeta_j}$, where: Φ' is distributed as Φ, conditional to $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_k)$, each Φ_{ζ_j} is a Poisson point process with intensity $\nu_{\zeta_j}(\mathrm{d}x) = \gamma k_\alpha(\zeta_j - x)\mathrm{d}x$, and*

$$\zeta_j \sim p_\zeta(v) = \frac{\gamma k_\alpha(x_j - v)\nu(\mathrm{d}v)}{m_\Phi(\mathrm{d}x_j)}$$

When $k = 1$, an equivalent result, stated in terms of spatial point patterns, is found in Møller (2003).

Proof. Let $\boldsymbol{x} = (x_1, \dots, x_k)$, then

$$\begin{aligned}
\mathbb{E}\left[\int_{\mathbb{X}^k} g(\boldsymbol{x}, \Phi - \sum_{j=1}^k \delta_{x_j})\Phi^k(\mathrm{d}\boldsymbol{x})\right] &= \mathbb{E}_\Lambda\left[\mathbb{E}_{\Phi|\Lambda}\left[\int_{\mathbb{X}^k} g(\boldsymbol{x}, \Phi)\Phi^k(\mathrm{d}\boldsymbol{x}) \mid \Lambda\right]\right] \\
&= \mathbb{E}_\Lambda\left[\int_{\mathbb{X}^k} \mathbb{E}_{\Phi|\Lambda}\left[g\left(x, \Phi\right) \mid \Lambda\right] \prod_{j=1}^k \nu_\Lambda(\mathrm{d}x_j)\right] \\
&= \mathbb{E}_\Lambda\left[\int_{\mathbb{X}} g'(\boldsymbol{y}, \Lambda)\Lambda^k(\mathrm{d}\boldsymbol{y})\right]
\end{aligned}$$

where $\boldsymbol{y} = (y_1, \dots, y_k)$ and

$$g'(\boldsymbol{y}, \Lambda) = \int_{\mathbb{X}^k} \mathbb{E}_{\Phi|\Lambda}\left[g\left(x, \Phi\right) \mid \Lambda\right] \prod_{j=1}^k \gamma k_\alpha(x_j - y_j)\mathrm{d}x_j$$

while the second equation follows from the Slyvniak-Mecke theorem. Again by Slyvniak-Mecke theorem, we have that

$$\begin{aligned}
\mathbb{E}\left[\int_{\mathbb{X}^k} g(\boldsymbol{x}, \Phi - \sum_{j=1}^k \delta_{x_j})\Phi^k(\mathrm{d}\boldsymbol{x})\right] &= \int_{\mathbb{X}^k} \mathbb{E}_\Lambda\left[g'(\boldsymbol{y}, \Lambda)\right] \prod_{j=1}^k \lambda(\mathrm{d}y_j) \\
= \int_{\mathbb{X}^k} \mathbb{E}_\Lambda & \left[\int_{\mathbb{X}^k} \mathbb{E}_{\Phi|\Lambda + \sum_{j=1}^k \delta_{y_j}}\left[g\left(\boldsymbol{x}, \Phi\right) \mid \Lambda\right] \prod_{j=1}^k \gamma k_\alpha(x_j - y_j)\mathrm{d}x_j\right] \prod_{j=1}^k \lambda(\mathrm{d}y_j) \\
= \int_{\mathbb{X}^k} \mathbb{E}_\Lambda & \left[\int_{\mathbb{X}^k} \mathbb{E}_{\Phi|\Lambda + \sum_{j=1}^k \delta_{y_j}}\left[g\left(\boldsymbol{x}, \Phi\right) \mid \Lambda\right] \prod_{j=1}^k \gamma k_\alpha(x_j - y_j)\lambda(\mathrm{d}y_j)\right] \prod_{j=1}^k \mathrm{d}x_j
\end{aligned}$$

where $\Phi|\Lambda + \sum_{j=1}^k \delta_{y_j}$ is a Poisson process with intensity $\nu_\Lambda(x)\mathrm{d}x + \sum_{j=1}^k \gamma k_\alpha(x - y_j)\mathrm{d}x$ and the last equality follows from Fubini's theorem. Focusing on the innermost integral, by multiplying and dividing by $\prod_{j=1}^k m_\Phi(x_j) = \gamma^k \int_{\mathbb{X}^k} \prod_{j=1}^k k_\alpha(x_j - z_j)\lambda(\mathrm{d}z_j)$, we have that we can introduce k auxiliary variables $\boldsymbol{\zeta} := (\zeta_1, \dots, \zeta_k)$ independently distributed as

$$\zeta_j \sim p_\zeta(y) = \frac{\gamma k_\alpha(x_j - y_j)\lambda(\mathrm{d}y_j)}{m_\Phi(\mathrm{d}x_j)}$$

such that

$$\begin{aligned}
\int_{\mathbb{X}^k} \mathbb{E}_{\Phi|\Lambda + \sum_{j=1}^k \delta_{y_j}}\left[g\left(\boldsymbol{x}, \Phi\right) \mid \Lambda\right] \prod_{j=1}^k \gamma k_\alpha(x_j - y_j)\lambda(\mathrm{d}y_j) = \\
\mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\Phi|\Lambda + \sum_{j=1}^k \delta_{\zeta_j}}\left[g(x, \Phi + \delta_x) \mid \Lambda\right]\right] M_\Phi^k(\mathrm{d}x).
\end{aligned}$$

Hence, we can recognize in the term $\mathbb{E}_\Lambda\left[\mathbb{E}_{\boldsymbol{\zeta}}\left[\mathbb{E}_{\Phi|\Lambda + \sum_{j=1}^k \delta_{x_j}}\left[g(x, \Phi + \delta_x) \mid \Lambda\right]\right]\right]$ the law of a Cox process with random intensity $\nu_{\Lambda,\zeta}(\mathrm{d}x) = \gamma \int_X k_\alpha(x - y)\left(\Lambda + \sum_{j=1}^k \delta_{\zeta_j}\right)(\mathrm{d}y)\ \mathrm{d}x$.

The proof follows by writing $\Phi|\Lambda + \sum_{j=1}^k \delta_{\zeta_j}$ as the sum of $k+1$ independent Poisson processes, the first one with random intensity $\nu_\Lambda(x)\mathrm{d}x = \int_{\mathbb{X}} \gamma k_\alpha(x - y)\Lambda(\mathrm{d}y)\mathrm{d}x$ and the remaining ones with random intensity $\nu_{\zeta_j} = \gamma k_\alpha(x - \zeta_j)$, $j = 1, \dots, k$.

□

Lemma 3.5. *Let $\Psi' = \sum_{j\geq 1}\delta_{X'_j,S'_j}$ where $S' \overset{iid}{\sim} H'$ and $\Phi' = \sum_{j\geq 1}\delta_{X'_j}$ is a shot-noise Cox point process with base intensity ρ'. Let*

$$\mu'(A) = \int_{A\times\mathbb{R}_+} s\Psi'(\mathrm{d}x\mathrm{d}s)$$

Then for any $f \geq 0$

$$\mathbb{E}\left[e^{-\int_{\mathbb{X}} f(x)\mu'(\mathrm{d}x)}\right] = \exp\left\{-\int_{\mathbb{X}} 1 - \exp\left(-\int_{\mathbb{X}}\gamma k_\alpha(x-y)\int_{\mathbb{R}_+} 1 - e^{-sf(x)}H'(\mathrm{d}s)\mathrm{d}x\right)\rho'(\mathrm{d}y)\right\}$$

Proof. By Lemma 3.3 we have that

$$\mathbb{E}\left[e^{-\int_{\mathbb{X}} f(x)\mu'(\mathrm{d}x)}\right] = \mathbb{E}\left[\exp\left\{\int_{\mathbb{X}}\log\left(\int_{\mathbb{R}_+} e^{-sf(x)}H'(\mathrm{d}s)\right)\Phi'(\mathrm{d}x)\right\}\right]$$

Then by the tower property of the expected value and exploiting the Lévy-Kintchine representation and Fubini's theorem, we have

$$\begin{aligned}\mathbb{E}\left[e^{-\int_{\mathbb{X}} f(x)\mu'(\mathrm{d}x)}\right] &= \mathbb{E}\left[\mathbb{E}\left[\exp\left\{\int_{\mathbb{X}}\log\left(\int_{\mathbb{R}_+} e^{-sf(x)}H'(\mathrm{d}s)\right)\Phi'(\mathrm{d}x)\right\}\mid\Lambda\right]\right]\\ &= \mathbb{E}\left[\exp\left\{-\int_{\mathbb{X}}\left(1-\int_{\mathbb{R}_+} e^{-sf(x)}H'(\mathrm{d}s)\right)\gamma\int_{\mathbb{X}} k_\alpha(x-y)\Lambda(\mathrm{d}y)\mathrm{d}x\right\}\right]\\ &= \mathbb{E}\left[\exp\left\{-\int_{\mathbb{X}}\int_{\mathbb{X}}\gamma k_\alpha(x-y)\int_{\mathbb{R}_+}\left(1-e^{-sf(x)}H'(\mathrm{d}s)\right)\mathrm{d}x\Lambda(\mathrm{d}y)\right\}\right]\\ &= \exp\left\{-\int_{\mathbb{X}} 1-\exp\left(-\int_{\mathbb{X}}\gamma k_\alpha(x-y)\int_{\mathbb{R}_+} 1-e^{-sf(x)}H'(\mathrm{d}s)\mathrm{d}x\right)\rho'(\mathrm{d}y)\right\}\end{aligned}$$

□

Lemma 3.6. *Let Φ be a shot-noise Cox process with kernel k_α and base intensity ρ. Define $\eta(x_1,\ldots,x_l) := \int\prod_{i=1}^{l} k_\alpha(x_i - v)\nu(\mathrm{d}v)$. Then*

$$M^k_\Phi(\mathrm{d}x_1\cdots\mathrm{d}x_k) = \gamma^k\sum_{j=1}^{k}\sum_{C_1,\ldots C_j\in(*)}\prod_{l=1}^{j}\eta(\boldsymbol{x}_{C_l})\mathrm{d}x_1\cdots\mathrm{d}x_k.$$

where $(*)$ denotes all the partition of k elements in j groups.

Proof. By Campbell's theorem:

$$\begin{aligned}M^k_\Phi(\mathrm{d}\boldsymbol{x}) &= \mathbb{E}\left[\prod_{i=1}^{k}\mathbb{E}\left[\Phi(\mathrm{d}x_i)\mid\Lambda\right]\right] = \gamma^k\mathbb{E}\left[\int_{\mathbb{X}^k}\prod_{i=1}^{k} k_\alpha(x_i-v_i)\Lambda(\mathrm{d}v_1)\cdots\Lambda(\mathrm{d}v_k)\right]\mathrm{d}x_1\cdots\mathrm{d}x_k\\ &= \gamma^k\int_{\mathbb{X}^k}\prod_{i=1}^{k} k_\alpha(x_i-v_i)M_{\Lambda^k}(\mathrm{d}v_1\cdots\mathrm{d}v_k)\mathrm{d}x_1\cdots\mathrm{d}x_k\end{aligned}$$

where M_{Λ^k} is the k-th moment measure of the Poisson point process Λ, which can be expressed as

$$M_{\Lambda^k}(\mathrm{d}v_1\cdots\mathrm{d}v_k) = \sum_{j=1}^{k}\sum_{C_1,\ldots C_j\in(*)}\prod_{l=1}^{j}\left[\nu(\mathrm{d}v_{C_{l1}})\prod_{m\in C_l}\delta_{v_{C_{l1}}}(v_{C_{lm}})\right]$$

where $(*)$ denotes all the partition of k elements in j groups. Then

$$M_{\Phi}^{k}(\mathrm{d}\boldsymbol{x}) = \gamma^{k} \sum_{j=1}^{k} \sum_{C_1,\dots C_j \in (*)} \int_{\mathbb{X}^k} \prod_{i=1}^{k} k_\alpha(x_i - v_i) \prod_{l=1}^{j} \left[\nu(\mathrm{d}v_{C_{l1}}) \prod_{m \in C_l} \delta_{v_{C_{l1}}}(v_{C_{lm}}) \right] \mathrm{d}x_1 \cdots \mathrm{d}x_k$$

observe that the integral over $\mathbb{X}^k$ has a nice interpretation of product of marginals of models of the kind $\boldsymbol{x}_{C_l} := (x_i : i \in C_l) \mid v_{C_{l1}} \overset{\text{iid}}{\sim} k_\alpha(\cdot \mid v_{C_{l1}})$, $v_{C_{l1}} \sim \rho$. Denoting with $\eta(\boldsymbol{x}_{C_l})$ the marginal distribution for the model, the integral can be expressed as $\prod_{l=1}^{j} \eta(\boldsymbol{x}_{C_l})$ leading to

$$M_{\Phi}^{k}(\mathrm{d}\boldsymbol{x}) = \gamma^{k} \sum_{j=1}^{k} \sum_{C_1,\dots C_j \in (*)} \prod_{l=1}^{j} \eta(\boldsymbol{x}_{C_l}) \mathrm{d}x_1 \cdots \mathrm{d}x_k.$$

□

3.G.2 Proof of Theorem 3.4

We are now ready to prove Theorem 3.4

Proof. Consider first the numerator in (3.9) as

$$\mathbb{E}\left[e^{-\int_X (f(x)+u)\mu^!(\mathrm{d}x)}\right] = \mathbb{E}\left[\exp\left\{\int_{\mathbb{X}} \log\left(\int_{\mathbb{R}_+} e^{-(f(x)+u)} H(\mathrm{d}s)\right) \Phi^!_{\mathbf{y}^*}(\mathrm{d}x)\right\}\right].$$

Recall now that, from Proposition 3.4

$$\Phi^!_{\mathbf{y}^*}(\mathrm{d}x) = \Phi' + \sum_{j=1}^{k} \Phi_{\zeta_j}$$

so that

$$\mathbb{E}\left[e^{-\int_X (f(x)+u)\mu^!(\mathrm{d}x)}\right] = \prod_{j=1}^{k} \mathbb{E}\left[e^{\int_{\mathbb{X}} \log\left(\int_{\mathbb{R}_+} e^{-(f(x)+u)s} H(\mathrm{d}s)\right) \Phi_{\zeta_j}(\mathrm{d}x)}\right] \times$$
$$\mathbb{E}\left[e^{\int_{\mathbb{X}} \log\left(\int_{\mathbb{R}_+} e^{-(f(x)+u)s} H(\mathrm{d}s)\right) \Phi'(\mathrm{d}x)}\right]$$

The denominator in (3.9) can be recovered in the previous expressions setting $f \equiv 0$. Therefore, we can evaluate a product of ratio of expectations. Let us consider the term involving Φ' first. Proceeding along the same lines of Lemma 3.5, we have

$$\begin{aligned}
&\mathbb{E}\left[e^{\int_{\mathbb{X}} \log\left(\int_{\mathbb{R}_+} e^{-(f(x)+u)s} H(\mathrm{d}s)\right) \Phi'(\mathrm{d}x)}\right] = \\
&\quad = \mathbb{E}_\Lambda\left[\mathbb{E}\left[e^{\int_{\mathbb{X}} \log\left(\int_{\mathbb{R}_+} e^{-(f(x)+u)} H(\mathrm{d}s)\right) \Phi'(\mathrm{d}x)} \mid \Lambda\right]\right] \\
&\quad = \mathbb{E}_\Lambda\left[\exp\left\{-\int_{\mathbb{X}} 1 - \left(\int_{\mathbb{R}_+} e^{-(f(x)+u)s} H(\mathrm{d}s)\right) \gamma \int_{\mathbb{X}} k_\alpha(x-y)\Lambda(\mathrm{d}y)\mathrm{d}x\right\}\right] \\
&\quad = \mathbb{E}_\Lambda\left[\exp\left\{-\int_{\mathbb{X}}\int_{\mathbb{X}} 1 - \left(\int_{\mathbb{R}_+} e^{-(f(x)+u)s} H(\mathrm{d}s)\right) \gamma k_\alpha(x-y)\mathrm{d}x\Lambda(\mathrm{d}y)\right\}\right] \\
&\quad = \exp\left\{-\int_{\mathbb{X}} 1 - \exp\left[-\int_{\mathbb{X}} 1 - \left(\int_{\mathbb{R}_+} e^{-(f(x)+u)s} H(\mathrm{d}s)\right) \gamma k_\alpha(x-y)\mathrm{d}x\right] \nu(\mathrm{d}y)\right\}
\end{aligned}$$

where the second equality follows from the Lévy-Khintchine representation for $\Phi' \mid \Lambda$, the third one by Fubini's theorem and the last one from the Lévy-Khintchine representation for Λ. When $f \equiv 0$ the expression reduces to

$$\mathbb{E}\left[e^{\int_{\mathbb{X}} \log\left(\int_{\mathbb{R}_+} e^{-us} H(\mathrm{d}s)\right) \Phi'(\mathrm{d}x)}\right] =$$

$$= \exp\left\{-\int_{\mathbb{X}} 1 - \exp\left[-\gamma \int_{\mathbb{R}_+} 1 - e^{-us} H(\mathrm{d}s)\right] \nu(\mathrm{d}y)\right\}$$

Taking the ratio of the two yields

$$\begin{aligned}
&\frac{\mathbb{E}\left[e^{\int_{\mathbb{X}} \log\left(\int_{\mathbb{R}_+} e^{-(f(x)+u)s} H(\mathrm{d}s)\right) \Phi'(\mathrm{d}x)}\right]}{\mathbb{E}\left[e^{\int_{\mathbb{X}} \log\left(\int_{\mathbb{R}_+} e^{-us} H(\mathrm{d}s)\right) \Phi'(\mathrm{d}x)}\right]} \\
&\qquad = \exp\Bigg\{ -\int_{\mathbb{X}} \exp\left[-\gamma \int_{\mathbb{R}_+} 1 - e^{-us} H(\mathrm{d}s)\right] - \\
&\qquad\qquad \exp\left[-\int_{\mathbb{X}} \gamma k_\alpha(x-y) \int_{\mathbb{R}_+} 1 - e^{-(f(x)+u)s} H(\mathrm{d}s)\mathrm{d}x\right] \nu(\mathrm{d}y)\Bigg\} \\
&\qquad = \exp\Bigg\{ -\int_{\mathbb{X}} \left[1 - \exp\left(-\int_{\mathbb{X}} \gamma k_\alpha(x-y) \int_{\mathbb{R}} e^{-sf(x)} e^{-us} H(\mathrm{d}s)\mathrm{d}x\right)\right] \\
&\qquad\qquad \times e^{-\gamma \int_{\mathbb{R}_+} 1 - e^{-us} H(\mathrm{d}s)} \nu(\mathrm{d}y)\Bigg\}
\end{aligned}$$

where we recognize the same expression in Lemma 3.5 by setting

$$H'(\mathrm{d}s) := \frac{e^{-us} H(\mathrm{d}s)}{\int_{\mathbb{R}_+} e^{-us} H(\mathrm{d}s)}$$

and $\rho'(\mathrm{d}y) = e^{-\gamma \int_{\mathbb{R}_+} 1 - e^{-us} H(\mathrm{d}s)} \nu(\mathrm{d}y)$.

Regarding the ratio involving one of the Φ_{ζ_j}, using the same techniques as above it is easy to show that

$$\begin{aligned}
&\frac{\mathbb{E}\left[e^{\int_{\mathbb{X}} \log\left(\int_{\mathbb{R}_+} e^{-(f(x)+u)s} H(\mathrm{d}s)\right) \Phi_{\zeta_j}(\mathrm{d}x)}\right]}{\mathbb{E}\left[e^{\int_{\mathbb{X}} \log\left(\int_{\mathbb{R}_+} e^{-us} H(\mathrm{d}s)\right) \Phi_{\zeta_j}(\mathrm{d}x)}\right]} = \\
&\qquad \mathbb{E}_{\zeta_j}\left[\exp\left(-\int_{\mathbb{X}} 1 - \int_{\mathbb{R}_+} e^{-f(x)s} H'(\mathrm{d}s) e^{-\gamma \int_{\mathbb{R}_+} 1 - e^{-us} H(\mathrm{d}s)} \lambda_{\zeta_j}(\mathrm{d}x)\right)\right]
\end{aligned}$$

where we recognize on the right hand side the Laplace transform of the random measure $\widetilde{\mu}_{\zeta_j} = \sum_{j \geq 1} S'_j \delta_{X'_j}$ where $S'_j \overset{\text{iid}}{\sim} H'(\mathrm{d}s)$ (defined above) and the point process $\Phi_{\zeta_j} = \sum_{j \geq 1} \delta_{X'_j}$ is a Poisson point process with random intensity $\lambda_{\zeta_j}(\mathrm{d}x) = \gamma e^{-\gamma \int_{\mathbb{R}_+} 1 - e^{-us} H(\mathrm{d}s)} k_\alpha(\zeta_j - x)\mathrm{d}x$ and

$$\zeta_j \sim p_\zeta(y) = \frac{\gamma k_\alpha(x_j - y)\lambda(\mathrm{d}y)}{m_\Phi(\mathrm{d}x_j)}$$

□

3.G.3 Marginal and Predictive distribution

We can take advantage of the hierarchical structure of the shot noise cox process to evaluate the predictive distribution in Theorem 3.3. In the following, we assume that $k_\alpha(x)$ integrates to one for simplicity. From Lemma 3.3

$$\begin{aligned}\mathbb{E}\left[e^{-\int_{\mathbb{X}} u \mu^!_{\boldsymbol{y}^*}(\mathrm{d}x)}\right] &= \mathbb{E}\left[\exp\left\{-\log\psi(u)\Phi^!(\mathbb{X})\right\}\right] \\ &= \mathbb{E}\left[\exp\left\{-\log\psi(u)\Phi_0(\mathbb{X})\right\}\right]\prod_{j=1}^{k}\mathbb{E}\left[\exp\left\{-\log\psi(u)\Phi_{\zeta_j}(\mathbb{X})\right\}\right]\end{aligned}$$

Observe that $\Phi_{\zeta_j}(\mathbb{X})$, conditionally to ζ_j is a Poisson random variable with parameter $\gamma\int_{\mathbb{X}} k_\alpha(x-\zeta_j)\mathrm{d}x = \gamma$, so that its law is independent of ζ_j. Hence

$$\mathbb{E}\left[\exp\left\{-\log\psi(u)\Phi_{\zeta_j}(\mathbb{X})\right\}\right] = \exp\left\{\gamma(\psi(u)^{-1}-1)\right\} \tag{3.31}$$

$\Phi_0(\mathbb{X})$ is distributed as a shot-noise cox process, so that conditionally to $\Lambda \sim \mathrm{PRM}(\rho)$ we have $\Phi_0(\mathbb{X}) \mid \Lambda \sim \mathrm{Poi}(\gamma\Lambda(\mathbb{X}))$. Letting $\lambda \equiv \int_{\mathbb{X}} \nu(x)\mathrm{d}x$:

$$\mathbb{E}\left[\exp\left\{-\log\psi(u)\Phi_0(\mathbb{X})\right\}\right] = \exp\left\{\lambda\left(e^{\gamma(\psi(u)^{-1}-1)}-1\right)\right\} \tag{3.32}$$

Combining Equations (3.32) and (3.31) accounts for the exponential term in Corollary 3.3, while the moment measure is as in Lemma 3.6.

Finally, observe that the ratio of expected values in Theorem 3.3 reduces to

$$\frac{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u\mu^!_{(\boldsymbol{y}^*,y)}(\mathrm{d}z)}\right]}{\mathbb{E}\left[e^{-\int_{\mathbb{X}} u\mu^!_{\boldsymbol{y}^*}(\mathrm{d}z)}\right]} = \exp\left\{\gamma(\psi(u)^{-1}-1)\right\}.$$

3.G.4 Details about the MCMC algorithm

Let $\Lambda = \sum_m \delta_{v_m}$ be the directing Poisson process. We introduce auxiliary latent variables t_h, one for each atom of the measure μ so that $x_h \mid \Lambda, t_h = k \sim k_\alpha(x - v_k)$.

1. Sample $u \sim \Gamma(n, T)$,

2. Consider the concatenation of all the atoms in the measure μ, i.e., $y^*_1, \ldots, y^*_k, x_{1,1}, \ldots, x_{1,n_1}, \ldots, x_{k,1}, \ldots, x_{k,n_k}, x'_1, \ldots, x'_{n'}$, where the first k correspond to the allocated points of support, the following n_1 correspond to the measure μ_{ζ_1} and so on, the last n' correspond to the measure μ' in Theorem 3.4. Denote them with $x_1, \ldots, x_K$ and analogously for $\gamma_1, \ldots, \gamma_K$ and $s_1, \ldots, s_K$. Sample the cluster allocations from a categorical distribution with weights

$$P(c_i = h \mid \text{rest}) \propto s_h f(y_i \mid x_h, \gamma_h)$$

   Relabel cluster allocation labels so that the k_a unique values in $\boldsymbol{c}$ are $\{1, \ldots, m\}$ and the mixture components analogously.

3. Sample the active part

   (a) Sample $s_h \sim p(s) \propto s^{n_h} e^{-us} h(s)$, $h = 1, \ldots, k_a$

   (b) Sample $x_h \sim p(x) \propto \prod_{i:c_i=h} f(y_i \mid x, \gamma_h) k_\alpha(x - v_{t_h})$ $h = 1, \ldots, k_a$

   (c) Sample $\gamma_h \sim p(\gamma) \propto \prod_{i:c_i=h} f(y_i \mid x_h, \gamma)\pi(\gamma)$

4. Sample the non-active part

   (a) For $h = 1, \dots, k$

      i. sample $\zeta_h \sim p_{\zeta_h}(v) \propto k_\alpha(x_h - v)\lambda(v)$

      ii. sample $x_{h,1}, \dots, x_{h,n_h}$ from a Poisson process with intensity λ'_{ζ_h} as in Theorem 3.4, set the corresponding variables $t_{h,1}, \dots, t_{h,n_h}$ equal to $k + h$

      iii. sample $x'_1, \dots, x'_{n_h}$ from a shot-noise Cox process with base intensity ρ' as in as in Theorem 3.4, set the corresponding variables $t'_1, \dots, t'_{n'}$ by the generative process.

   (b) For all the x_h's simulated above, sample $s_h \sim p(s) \propto e^{-us}h(s)$, $h = k_a + 1, \dots$

   (c) For all the x_h's simulated above, sample $\gamma_h \sim \pi(\gamma)$

5. Sample the latent Poisson process

   (a) Sample the latent variables t_h from a categorical distribution over all the atoms of Λ with weights $P(t_h = k \mid \text{rest}) \propto k_\alpha(x_h - v_h)$. Relabel the t_h's and the atoms of Λ so that the unique values in the t_h's are the first ones.

   (b) Sample the latent centers v_k from $p(v) \propto \prod_{h:t_h=k} k_\alpha(x_h - v)\lambda(v)$

4. Clustering high dimensional data via latent repulsive mixtures

In this chapter, we extend the construction of repulsive mixture models in Chapter 2 to the high-dimensional setting, that is, when the data are p dimensional and p is large compared to the sample size. We extend the Lamb model proposed in Chandra et al. (2020), by assuming an *anisotropic* repulsive mixture for the prior of the cluster centers, which is essential to obtain well separated clusters of observations. In particular, we propose a general construction for anisotropic determinantal point processes, which generalizes the construction in Lavancier et al. (2015), providing easy-to-check conditions for the existence of the process as well as explicit formulas for its spectral density, which is essential for simulations.

4.1 Introduction

High-dimensional data are routinely collected in a vast number of applicative fields, such as genomics (see Kiselev et al., 2019, for a review on single-cell data), text mining (Blei et al., 2003), and ecology (Dunstan et al., 2013). In this chapter, we consider observations $y_1, \dots, y_n \in \mathbb{R}^p$, where p is large compared to n. Cluster analysis might be particularly useful for such high-dimensional datasets, as it provides a straightforward procedure for exploring the data by exploiting the *latent* structure arising from similar observations. Moreover, it can be an important preprocessing step for subsequent analyses.

Bayesian model-based clustering is appealing in the large p setting, as it implicitly quantifies uncertainty. Although a variety of models have been proposed in the literature (Neal, 2003; Teh et al., 2007; Duan and Dunson, 2021; Natarajan et al., 2021) Bayesian mixtures are the most direct model for model-based clustering; see Fruhwirth-Schnatter et al. (2019) for a recent review. In mixture models, it is assumed that data are generated from m (either random or fixed) homogeneous populations. Typically, each population is assumed to be suitably modelled via a parametric density $f_\theta(\cdot)$ for some parameter $\theta \in \Theta$. Weights $\boldsymbol{w} = (w_1, \dots, w_m)$ ($w_h \geq 0$, $\sum_{h=1}^m w_h = 1$) specify the relative frequency of each population. Summing up, the conditional distribution of data, given parameters, under the mixture model takes the form

$$y_1, \dots, y_n \mid \boldsymbol{w}, \boldsymbol{\theta} \overset{\text{iid}}{\sim} p(\cdot) = \sum_{h=1}^{m} w_h f_{\theta_h}(\cdot) \tag{4.1}$$

Under the Bayesian approach, suitable priors are assumed for $\boldsymbol{w}$, $\boldsymbol{\theta} = (\theta_1, \dots, \theta_m)$, and m.

The poor performance of Bayesian mixtures in the large-p setting is well-known. The issue is not only due to the poor scalability of the algorithms for posterior inference (see, e.g., Malsiner-Walli et al., 2016; Celeux et al., 2019), but, as shown in Chandra et al. (2020) has its roots in the choice of the mixture kernel f_θ in (4.1). Specifically, Theorem 1 together with Corollaries 1 and 2 in Chandra et al. (2020) entail that the popular Gaussian distribution leads to inconsistent clustering when $p \to \infty$: if the covariance matrix is cluster-specific, then with probability one all the observations will be clustered into a different singleton cluster, while if the covariance matrix is shared through all the clusters, only one cluster is detected.

To overcome such limitations, Chandra et al. (2020) propose to use LAMB, a class of factor analytic models (see Section 4.2 below) where clustering is performed at a latent level, on $d \ll p$ dimensional latent parameters. Essentially, LAMB assumes that data $y_i \in \mathbb{R}^p$ lie close to a hyperplane of dimension d, with the addition of a Gaussian error. Such a hypothesis, although probably unrealistic, is common to all factor analytic models. Despite possible misspecification, due to the high degree of interpretability inherited by the linear structure (in fact, it is possible to interpret the matrix of factor loadings as well as the scores), factor models enjoy a large popularity in several fields such as genomics (Lucas et al., 2006; Carvalho et al., 2008), econometrics (Geweke and Zhou, 1996) and ecology (Schiavon et al., 2022).

Although LAMB models overcome the cluster inconsistency commonly caused by inappropriate choices of f_θ in the large-p setting, we still may expect the clustering to be inconsistent for two reasons. First, the LAMB model in Chandra et al. (2020) assumes a Dirichlet process (DP) prior for the latent variables and cluster inconsistency under the broader class of Pitman–Yor processes has been established in Miller and Harrison (2014b). The second reason has to deal with the impact of misspecification in mixture models, as recently investigated in Cai et al. (2021). Indeed, they show that, even if one replaced the inconsistent DP with a finite mixture model with a random number of components (see Miller and Harrison, 2018, and the references therein) or with an overfitted mixture (Rousseau and Mengersen, 2011), the model would tend to overestimate the number of clusters. In fact, in case of misspecified likelihoods, the number of components would need to be larger than the number of populations in the data in order to well approximate the data-generating density.

In general, when a mixture model is not well specified, we can identify a trade-off between the accuracy of cluster detection and density estimate: better density estimates necessarily yield poorer cluster estimates. As shown in Cai et al. (2021), traditional mixture models tend to favor density over cluster estimates. Repulsive mixture models (Beraha et al., 2022, and the references therein) are an attempt to reverse the trade-off in favor of better cluster estimates: by encouraging well-separated components, repulsive mixtures usually have poorer density estimates, but do not overestimate the number of clusters.

In the present chapter, we propose APPLAM: Anisotropic (repulsive) Point Process LAtent Mixture. We combine the idea behind LAMB with repulsive mixture models, where a repulsive point process is assumed as prior for the cluster-specific parameters. We argue that in order to have well separated clusters of data, it is not sufficient to have well separated clusters at the latent level, but the repulsion should take into account also the factor analytic model that links the latent variables to the observations. To this end, we propose to employ an anisotropic determinantal point process (DPP) as the prior distribution for the cluster specific parameters, where the anisotropy is driven by the matrix of factor loadings. We derive a general construction of anisotropic DPPs inducing the desired repulsion. We show existence conditions for our class of DPPs that resemble the ones in Lavancier et al. (2015) and further provide an explicit expression for the spectral density of the DPP, which is essential for simulation purposes. Moreover, we design an efficient block Gibbs sampler for posterior simulations.

The rest of the chapter is organized as follows. Section 4.2 contains a concise introduction to repulsive mixture models and the main results concerning the general construction of anisotropic DPPs. Section 4.3 gives details of the MCMC sampling, with particular emphasis on the update of the matrix of factor loadings. In Section 4.4 we provide two simulation studies comparing our model and the Lamb model of Chandra et al. (2020), showing how, under model misspecification, our model results in more robust as well as more accurate posterior inference. The Appendix contains the proofs for our theoretical results, measure theoretic details, and further plots and tables about the simulations.

4.2 Methodology

4.2.1 Bayesian clustering via latent mixtures

Let $y_1, \ldots, y_n \in \mathbb{R}^p$, $\Lambda \in \mathbb{R}^{p\times d}$ be the matrix of factor loadings, $\eta_1, \ldots, \eta_n \in \mathbb{R}^d$ be a set of latent factors, and $\Sigma = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_p^2)$ $(\sigma_j > 0)$ be a diagonal covariance matrix. Let $\mathcal{N}_p$ denote the p–dimensional Gaussian distribution. As in Chandra et al. (2020), we assume the following LAMB model

$$\begin{aligned} y_i \mid \eta_i, \Lambda, \Sigma &\overset{\text{ind}}{\sim} \mathcal{N}_p(\Lambda\eta_i, \Sigma), & i = 1, \ldots, n \\ \eta_i \mid \boldsymbol{w}, \boldsymbol{\theta} &\overset{\text{iid}}{\sim} p(z) = \sum_{h=1}^m w_h f_{\theta_h}(z), & i = 1, \ldots, n, \end{aligned} \tag{4.2}$$

where f_{θ_h} is a generic probability density function on $\mathbb{R}^d$. The prior for $\boldsymbol{w} = (w_1, \ldots, w_m)$, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_m)$, m, Λ and the σ_j's will be specified in Section 4.2.3 below.

Introducing a set of latent cluster indicator variables c_i such that $P(c_i = h \mid \boldsymbol{w}) = w_h$, we can equivalently state the prior for the η_i's in (4.2) as

$$\eta_i \mid c_i = h, \boldsymbol{\theta} \overset{\text{ind}}{\sim} f_{\theta_h}, \qquad i = 1, \ldots, n. \tag{4.3}$$

Therefore, a cluster model is induced among the y_i's through the latent variables η_i's. In particular, y_i and y_j belong to the same cluster if η_i and η_j do, that is, if $c_i = c_j$.

A meaningful and interpretable clustering is obtained (a posteriori) if the observations belonging to different clusters are well separated. Repulsive mixture models encourage well separated clusters by assuming a prior for the cluster centers that favors regular (i.e., well separated) point configurations. For instance, if the θ_h's in (4.2) were cluster centers, we could easily force that $P\left(\|\theta_h - \theta_k\| > \delta\right) = 1$ for any user-defined δ by assuming that $\{\theta_1, \ldots, \theta_m\}$ is distributed as an hardcore point process (Møller and Waagepetersen, 2004). Alternatively, as in Quinlan et al. (2020); Xie and Xu (2019); Beraha et al. (2022) we could have a "softer" control over the distance by assuming a pairwise interaction point process. This means that, the point pattern $\boldsymbol{\theta} := \{\theta_1, \ldots, \theta_m\}$ has a density with respect to the unit-rate Poisson point process (defined on a suitable space) given by

$$p(\{\theta_1, \ldots, \theta_m\}) = \frac{1}{Z} \prod_{j=1}^m \phi_1(\theta_j) \prod_{1\le h<k\le m} \phi_2(\|\theta_h - \theta_k\|) \tag{4.4}$$

where ϕ_1 is bounded, ϕ_2 is nondecreasing, and Z is a normalization constant that is usually intractable. See Daley and Vere-Jones (2008) and Møller and Waagepetersen (2004) for the definition of density with respect to a Poisson point process.

In the LAMB setting, this prior choice would ensure that different clusters are associated with well-separated latent scores η_i' s, but is this enough to ensure well-separated clusters of data y_i's? Let us now consider the case where f_θ is the d–dimensional Gaussian distribution and $\theta_h = (\mu_h, \Delta_h)$ where $\mu_h \in \mathbb{R}^d$ is the mean vector and Δ_h is a $d \times d$ symmetric and positive definite covariance matrix. By the properties of the Gaussian distribution, we have that

$$\{y_i : c_i = h\} \mid \boldsymbol{c}, \boldsymbol{\mu}, \boldsymbol{\Delta}, \boldsymbol{\Lambda} \overset{\text{iid}}{\sim} \mathcal{N}_p(\Lambda\mu_h, \Lambda\Delta_h\Lambda^\top).$$

Hence, it is clear that it is not sufficient to encourage a priori that $\|\mu_h - \mu_k\|$ is large to obtain well separated clusters of datapoints, as the distance between cluster centers is $\|\Lambda\mu_h - \Lambda\mu_h\|$. To this end, we could easily modify (4.4) as

$$p(\{\mu_1, \ldots, \mu_m\} \,|\, \Lambda) = \frac{1}{Z} \prod_{j=1}^{m} \phi_1(\mu_j) \prod_{1 \leq h < k \leq m} \phi_2(\|\Lambda\mu_h - \Lambda\mu_k\|), \tag{4.5}$$

where the normalizing constant Z depends on ϕ_1, ϕ_2 and, most importantly, on Λ. Note that $\|\Lambda\mu_h - \Lambda\mu_k\|^2 = (\mu_h - \mu_k)^\top(\Lambda^\top\Lambda)(\mu_h - \mu_k) := \|\mu_h - \mu_k\|^2_{\Lambda^\top\Lambda}$. Therefore, we are essentially considering (4.4) under a different metric on $\mathbb{R}^d$, that is, a weighted Euclidean metric.

Beraha et al. (2022) discuss how to sample the parameters in any specific choice of ϕ_1 and ϕ_2 by means of the exchange algorithm (Møller et al., 2006; Murray et al., 2006). This requires sampling from the distribution of $\boldsymbol{\mu}$ (via a perfect simulation algorithm), and it is shown that this is actually feasible when the number of components in the mixture is small. Essentially, the exchange algorithm uses a random-walk proposal in a Metropolis-Hastings move on an extended parameter space, where an auxiliary variable is introduced to get rid of the ratio of intractable normalizing constants in the acceptance rate.

Note that, in Beraha et al. (2022), only a single one-dimensional parameter (appearing in the expression of ϕ_1) needs to be updated via the exchange algorithm. In our case, instead, (4.5) depends on the high-dimensional parameter Λ. Our preliminary investigation showed that updating Λ using the exchange algorithm results in extremely poor mixing due to the high-dimensionality of the matrix Λ. Unfortunately, the normalizing constant Z in the density of $\boldsymbol{\mu}$ makes it impossible to employ gradient-based sampling algorithms.

In the rest of the chapter, as a prior for $\{\mu_1, \ldots, \mu_h\}$, we propose an anisotropic determinantal point process (DPP), defined in the next section, where an analytical expression of the normalizing constant Z is found. This allows us to employ the popular Metropolis-adjusted Langevin algorithm when sampling from the full conditional of Λ, leading to better MCMC chains even when p is large. Furthermore, we provide an analytical expression for the gradient of (the logarithm of) the DPP density with respect to Λ and show that it is numerically cheap to compute, especially when compared to the gradients computed via automatic differentiation algorithms.

4.2.2 A general construction for anisotropic DPPs

Let Φ be a point process on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$. We will consider Φ as a random point configuration $\Phi \equiv \{\mu_1, \ldots, \mu_m\} \subset \mathbb{R}^d$ or as a random counting measure $\Phi(B) = \sum_{\mu_j \in \Phi} \mathbb{1}[\mu_j \in B]$ depending on the convenience.

Determinantal point processes (DPPs) are usually defined in terms of their m-th factorial moment measures (Macchi, 1975; Lavancier et al., 2015; Baccelli et al., 2020). Briefly, the m-th factorial measure of Φ is defined as

$$\Phi^{(m)}(B_1 \times \cdots \times B_m) = \sum_{\mu_1, \ldots, \mu_m \in \Phi}^{\neq} \mathbb{1}[\mu_1 \in B_1] \cdots \mathbb{1}[\mu_m \in B_m]$$

for measurable $B_1, \ldots, B_m \subset \mathbb{R}^d$. The summation is intended over all m-tuples of pairwise different points in Φ. The m-th factorial moment measure simply defined as $M_{\Phi^{(m)}}(B_1 \times \cdots \times B_m) = \mathbb{E}\left[\Phi^{(m)}(B_1 \times \cdots \times B_m)\right]$, where the expectation is with respect to Φ.

In order to define a DPP, let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{C}$ be a continuous covariance kernel. Then Φ is a DPP on $\mathbb{R}^d$ if, for all $m = 1, 2, \ldots$, its m-th factorial moment measure has a density (with respect to the m-folded product of the d-dimensional Lebesgue measure) which equals

$$\rho^m(\mu_1, \dots, \mu_m) = \det\{K(\mu_h, \mu_k)\}_{h,k=1,\dots,m}, \qquad \mu_1, \dots, \mu_m \in \mathbb{R}^d.$$

By Mercer's theorem $K(x, y) = \sum_{j\geq 1} \gamma_j \xi_j(x)\overline{\xi}_j(y)$ where the ξ_j's form an orthonormal basis for $L^2(\mathbb{R}^d; \mathbb{C})$ of complex-valued functions and the γ_j's are a summable nonnegative sequence. Then, existence of a DPP with kernel K is equivalent to $\gamma_j \leq 1$ for all j, see Macchi (1975). When restricted to a compact $S \subset \mathbb{R}^d$, Φ is still a DPP with kernel K (but restricted to $S \times S$). In particular, if $\gamma_j < 1$ for all j, Φ has a density with respect to the unit rate Poisson point process on S given by

$$p(\{\mu_1, \dots, \mu_m\}) = e^{|S|-D} \det\{C(\mu_h, \mu_k)\}_{h,k=1,\dots,m}, \qquad \mu_1, \dots, \mu_m \in S.$$

where $C(x, y) = \sum_{j\geq 1} \gamma_j/(1-\gamma_j)\xi_j(x)\overline{\xi}_j(y)$, $|S| = \int_S \mathrm{d}x$, and $D = -\sum_{j\geq 1} \log(1-\gamma_j)$. See Lavancier et al. (2015) for a proof of such results.

It is clear that analytic expressions for γ_j are crucial for inferential purposes. Following the so-called "spectral approach" by Lavancier et al. (2015), Bianchini et al. (2020) and Beraha et al. (2022) assume $K(x, y) = K_0(x-y)$. Instead of modelling K, they fix the ξ_j's as the Fourier basis and assume a parametric model for the γ_j's. This approach ensures the positive-definitiness of K and the existence of the DPP density, but is somewhat limited. In fact, in Bianchini et al. (2020) and Beraha et al. (2022), K is a stationary and isotropic function, i.e., $K(x, y) = K_0(\|x-y\|)$. In particular, isotropy is in opposition with our goal of forcing repulsion across the $\Lambda\mu_h$'s. Here below, we provide a general construction for stationary anisotropic DPPs, providing explicit expression for the Fourier transform of its kernel K_0, and easy-to-check conditions that guarantee the existence of the DPP.

Theorem 4.1. *Let Λ be a (fixed) $p \times d$ real matrix with full rank. Let W be a strictly positive random variable and let $h(y)$ be the marginal density of the random variable Y defined as*

$$Y \mid W \sim \mathcal{N}_d(0, W(\Lambda^T\Lambda)^{-1}) \tag{4.6}$$

Let $K_0(x) = \rho h(x)/h(0)$ for $x \in \mathbb{R}^d$ and $\rho > 0$. Then there exists a DPP Φ on $\mathbb{R}^d$ with kernel $K(x, y) = K_0(x-y)$ for $\rho \leq \rho_{\max}$ defined as

$$\rho_{\max} = \frac{|\Lambda^T\Lambda|^{\frac{1}{2}}}{(2\pi)^{\frac{d}{2}}}\,\mathbb{E}\big[W^{-\frac{d}{2}}\big], \tag{4.7}$$

and

$$K_0(x) = \frac{\rho}{\mathbb{E}\big[W^{-\frac{d}{2}}\big]}\,\mathbb{E}\left[W^{-\frac{d}{2}}\exp\left(-\frac{||\Lambda x||^2}{2W}\right)\right], \qquad x \in \mathbb{R}^d. \tag{4.8}$$

If $\varphi(x) = \mathcal{F}(K_0)(x)$ denotes the Fourier transform of K_0, we have that

$$\varphi(x) = \frac{\rho}{h(0)}\,\mathbb{E}\big[\exp\big(-2\pi^2 W x^T(\Lambda^T\Lambda)^{-1}x\big)\big], \qquad x \in \mathbb{R}^d \tag{4.9}$$

Moreover, for any compact $S \subset \mathbb{R}^d$, the restriction of Φ to S admits a density with respect to the unit rate Poisson point process on S if $\rho < \rho_{\max}$.

The parameter ρ in Theorem 4.1 is the intensity of the process, i.e., it controls the distribution of the number of points in the process. In particular, the expected number of points is equal to ρ. The condition $\rho < \rho_{\max}$ translates into a well-known trade-off between the intensity of the process and the repulsiveness (Lavancier et al., 2015). Observe that the DPP defined in Theorem 4.1 matches our desiderata of introducing repulsion among $\{\Lambda x_1, \dots, \Lambda x_m\}$ instead of among $\{x_1, \dots, x_m\}$. The specific shape of the described repulsion is determined by the appropriate choice of the random variable $W \sim p(w)$ as described below. Let us remark that explicit knowledge of the Fourier transform is essential for simulation purposes, as one typically approximates the density of the DPP using φ as described in Section 4.3.1.

In the rest of the chapter, we consider two specific choices for the random variable W in Theorem 4.1, leading to anisotropic counterparts of the Gaussian and Whittle-Matern DPPs discussed in Lavancier et al. (2015), referred to as Gaussian-like and Whittle-Matern-like DPPs in the following. Other choices of W are possible, but may lead to expressions for $\varphi(x)$ that are not available in closed form. In this case, $\varphi(x)$ could be approximated via Monte Carlo integration.

Corollary 4.1. *Using the same notation as in Theorem 4.1, let W be a degenerate random variable defined as*

$$W = |\Lambda^T\Lambda|^{\frac{1}{d}} c^{-\frac{2}{d}}, \qquad c > 0,$$

where Λ is fixed. Then the kernel K_0, its Fourier transform $\varphi = \mathcal{F}(K_0)$ and $\rho_{\max}$ equal respectively

$$\begin{aligned} K_0(x) &= \rho \exp\left(-\frac{||\Lambda x||^2}{2|\Lambda^T\Lambda|^{\frac{1}{d}}\, c^{-\frac{2}{d}}}\right), & x \in \mathbb{R}^d \\ \varphi(x) &= \rho\, \frac{(2\pi)^{\frac{d}{2}}}{c} \exp\left(-2\pi^2 |\Lambda^T\Lambda|^{\frac{1}{d}} c^{-\frac{2}{d}} x^T (\Lambda^T\Lambda)^{-1} x\right), & x \in \mathbb{R}^d \qquad (4.10) \\ \rho_{\max} &= \frac{c}{(2\pi)^{d/2}} \end{aligned}$$

To show the effect of Λ, we consider the pair correlation function (PCF, Lavancier et al., 2015)

$$g(\mu_1, \mu_2) = 1 - \frac{K_0(\mu_1,\mu_2)K_0(\mu_2,\mu_1)}{K_0(\mu_1,\mu_1)K_0(\mu_2,\mu_2)} = 1 - \exp\left(-\frac{||\Lambda x||^2}{2|\Lambda^T\Lambda|^{\frac{1}{d}}\, c^{-\frac{2}{d}}}\right)^2, \quad \mu_1, \mu_2 \in \mathbb{R}^d$$

and set $p = d = 2$ for visual purposes. Figure 4.2.1 shows the PCFs of two Gaussian-like DPPs with different $\Lambda \in \mathbb{R}^{2\times 2}$. In the left panel, Λ has eigenvectors $e_1 = (1, 0)^T$, $e_2 = (0, 1)^T$ and eigenvalues $\lambda_1 = 1$, $\lambda_2 = \lambda$, which induces stronger repulsion along the horizontal axis than along the vertical one. In the right panel, Λ has eigenvectors $e_1 = \sqrt{2}/2 \cdot (1, 1)^T$, $e_2 = \sqrt{2}/2 \cdot (-1, 1)^T$ and eigenvalues $\lambda_1 = 1$, $\lambda_2 = \lambda$, which induces stronger repulsion along the bisector of the first quadrant than along the orthogonal direction.

Corollary 4.2. *Using the same notation of Theorem 4.1, let*

$$W \sim \text{Gamma}\left(\nu + \frac{d}{2}, \frac{1}{2|\Lambda^T\Lambda|^{\frac{1}{d}}\alpha^2}\right), \qquad \nu, \alpha > 0,$$

where Λ is fixed. Then the kernel K_0, its Fourier transform $\varphi = \mathcal{F}(K_0)$ and $\rho_{\max}$ equal respectively

$$\begin{aligned} K_0(x) &= \rho\, \frac{2^{1-\nu}}{\Gamma(\nu)} \left\| \frac{\Lambda x}{\alpha\, |\Lambda^T\Lambda|^{\frac{1}{2d}}} \right\|^{\nu} K_\nu\left(\left\| \frac{\Lambda x}{\alpha\, |\Lambda^T\Lambda|^{\frac{1}{2d}}} \right\| \right), & x \in \mathbb{R}^d \\ \varphi(x) &= \rho\, \frac{\Gamma(\nu + \frac{d}{2})}{\Gamma(\nu)} \frac{(2\sqrt{\pi}\alpha)^d}{\left(1 + 4\pi^2\alpha^2 |\Lambda^T\Lambda|^{\frac{1}{d}}\, x^T (\Lambda^T\Lambda)^{-1} x\right)^{\nu + \frac{d}{2}}}, & x \in \mathbb{R}^d \qquad (4.11) \\ \rho_{\max} &= \frac{\Gamma(\nu)}{\Gamma(\nu + \frac{d}{2})\, (2\sqrt{\pi}\alpha)^d} \end{aligned}$$

where K_ν is the modified Bessel function of the second kind.

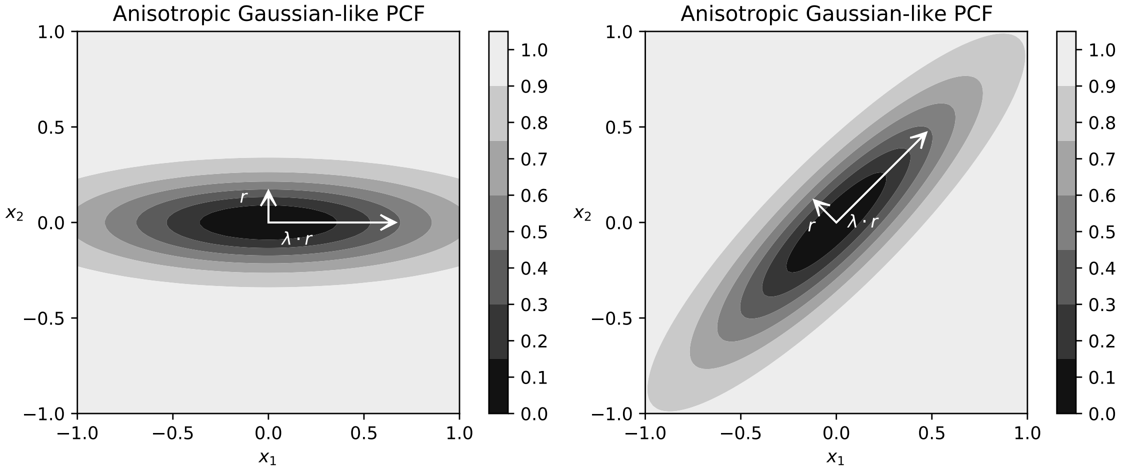


Figure 4.2.1: Pair correlation function of two Gaussian-like DPPs, one showing strong repulsion along the horizontal direction (left plot) and one showing strong repulsion along the bisector of the first quadrant (right plot).

In the following, we will use notation $\Phi \sim \mathrm{DPP}(\rho, \Lambda, K_0; S)$ to denote the law of a DPP on S with intensity ρ, kernel K_0 and anisotropy driven by Λ. Although the intensity and Λ are implicitly appearing inside the definition of K_0, we make it explicit in our notation to stress the importance of the parameters ρ and Λ.

4.2.3 APPLAM

The APPLAM model assumes likelihood (4.2), where f_θ is the d-dimensional Gaussian distribution with parameters $\theta = (\mu, \Delta)$. We complete prior specification as follows. First, we assume an anisotropic DPP prior for the cluster centers of the latent mixture, that is

$$\{\mu_1, \dots, \mu_m\} \,|\, \Lambda \sim \mathrm{DPP}(\rho, \Lambda, K_0; S)$$

where K_0 is either the Gaussian-like or the Whittle-Matérn-like kernel as defined in Corollaries 4.1 and 4.2. The choice of the compact set S is discussed in Section 4.3.1. The number of components m in the mixture (4.2) is random as well. In particular, note that $P(m = 0) > 0$ under the DPP prior. This may not be a concern in practice since a posteriori we always get $P(m = 0 \,|\, \boldsymbol{y}) = 0$. However, to have a well defined model we condition on $m > 0$. For a DPP, $P(m = 0) = 1 - e^{-D}$ so that we can assume the point process density

$$\begin{aligned} p(\{\mu_1, \dots, \mu_m\} \,|\, \Lambda) &= f_{\mathrm{DPP}}(\boldsymbol{\mu} \,|\, \rho, \Lambda, K_0; S) \\ &= \frac{e^{|S|-D}}{1 - e^{-D}} \det\{C(\mu_h, \mu_k)\}_{h,k=1,\dots,m}, \qquad m \geq 1,\, \mu_1, \dots, \mu_m \in S \end{aligned} \tag{4.12}$$

and $p(\emptyset \,|\, \Lambda) = 0$. Conditional to $\{\mu_1, \dots, \mu_m\}$ (and in particular only to m) we assume

$$\begin{aligned} w_1, \dots, w_m \,|\, m &\sim \mathrm{Dirichlet}(\alpha, \dots, \alpha) \\ \Delta_1, \dots, \Delta_m \,|\, m &\overset{\mathrm{iid}}{\sim} \mathrm{IW}_d(\nu_0, \Psi_0) \end{aligned} \tag{4.13}$$

where $\mathrm{IW}_d(\nu_0, \Psi_0)$ denotes the d-dimensional inverse Wishart distribution, with $\nu_0 > d-1$ degrees of freedom and mean $\Psi_0/(\nu_0 - d - 1)$.

As in Chandra et al. (2020) we assume that the diagonal elements σ_j^2 of Σ are independent and

$$\sigma_j^2 \overset{\mathrm{iid}}{\sim} \text{inv-Ga}(a_\sigma, b_\sigma), \qquad j = 1, \dots, p \tag{4.14}$$

Finally, a Dirichlet-Laplace prior with parameter a is assumed for Λ, that is, denoting with λ_{jh} the elements of Λ,

$$\begin{aligned} \lambda_{jh} \mid \phi, \tau, \psi &\overset{\text{ind}}{\sim} \mathcal{N}(0, \psi_{jh}\phi_{jh}^2\tau^2), \quad j = 1, \dots, p\, h = 1, \dots, d \\ vec(\phi) \sim \text{Dir}(a, \dots, a), \quad \psi_{jh} &\overset{\text{iid}}{\sim} \text{Exp}(1/2), \quad \tau \sim \text{Gamma}(pda, 1/2) \end{aligned} \tag{4.15}$$

where, for any $p \times d$ real matrix A, $vec(A)$ denotes the real vector of dimension $p \times d$ such that $vec(A)_{p(h-1)+j} = (A)_{j,h}$.

4.3 Posterior Simulation

4.3.1 Approximation of the DPP density

The DPP density in (4.12) cannot be evaluated in closed form due to the infinite series appearing in the expression of D and C. We follow Lavancier et al. (2015) and approximate it as follows. First, consider the case $S = [-1/2, 1/2]^d$. Let $\mathbb{Z}_N = \{-N, \dots, N\}$ for $N > 0$, for $m \geq 1$, $\mu_1, \dots, \mu_m \in S$, we define

$$\begin{aligned} f_{\text{DPP}}^{\text{app}}(\boldsymbol{\mu} \mid \rho, \Lambda, K_0; S) &= \frac{e^{|S| - D^{\text{app}}}}{1 - e^{-D^{\text{app}}}} \det\{C^{\text{app}}(\mu_h, \mu_k)\}_{h,k=1,\dots,m} \\ C^{\text{app}}(x, y) &= \sum_{k \in \mathbb{Z}_N^d} \frac{\varphi(k)}{1 - \varphi(k)} \exp\left(2\pi i \langle k, x - y \rangle\right), \qquad x, y \in S \\ D^{\text{app}} &= -\sum_{k \in \mathbb{Z}_N^d} \log(1 - \varphi(k)). \end{aligned} \tag{4.16}$$

The truncation level N controls both the quality of the approximation and the upper bound of N^d for the total number of points m in the DPP (see Equation (2.10) in Lavancier et al., 2015). We found that for Bayesian mixture modelling, small levels of truncation like $N = 3, 5$ produce satisfactory results. This is likely because the use of repulsive priors favors small values of m.

Finally, to approximate the density of DPPs defined on a hyperrectangular region R different from $[-1/2, 1/2]^d$, it is sufficient to consider an affine transformation $T : R \to S$ and perform a change of variable. In Beraha et al. (2022), R is fixed as the smallest hyperrectangle containing all the observations in an empirical Bayes fashion. Note that this procedure implicitly introduces anisotropy on the resulting point process if R is not a hypersquare. Therefore, we argue that R should be a hypersquare so that

$$\begin{aligned} &f_{\text{DPP}}^{\text{app}}(\boldsymbol{\mu} \mid \rho, \Lambda, K_0; R) = \\ &\qquad |R|^{-m} \frac{e^{|S| - D^{\text{app}}}}{1 - e^{-D^{\text{app}}}} \det\{C^{\text{app}}(T\mu_h, T\mu_k)\}_{h,k=1,\dots,m}, \qquad m \geq 1,\ \mu_1, \dots, \mu_m \in R \end{aligned}$$

In the rest of the chapter, we always fix $R = [-50, 50]^d$. Note that marginally, each μ_j is uniformly distributed on R. Numerical simulations show that posterior inference is robust with respect to R.

4.3.2 Gibbs sampling algorithm

Prior formulation in (4.13) is not amenable for posterior inference, as the sum-to-one constraint on $\boldsymbol{w}$ leads to complex split-merge reversible jump moves with poor mixing of the chain. As in Beraha et al. (2022) we consider the prior for $\boldsymbol{w}$ as the normalization of independent Gamma-distributed random variables, i.e.

$$\boldsymbol{w} = \left(\frac{S_1}{T}, \dots, \frac{S_m}{T}\right), \quad T = \sum_{h=1}^{m} S_h, \quad S_h \overset{\text{iid}}{\sim} \text{Ga}(\alpha, \beta) \tag{4.17}$$

Conditional to $\boldsymbol{c}$ we consider $\boldsymbol{\mu} = \mu^{(a)} \cup \mu^{(na)}$, $\boldsymbol{S} = [\boldsymbol{S}^{(a)}, \boldsymbol{S}^{(na)}]$ and $\boldsymbol{\Delta} = [\boldsymbol{\Delta}^{(a)}, \boldsymbol{\Delta}^{(na)}]$ into allocated and non-allocated components (denoted with the (a) and (na) superscript, respectively). Combining the likelihood (4.2) with prior assumptions (4.12)-(4.15), we can see that the joint distribution of data and parameters has a density, whose expression is reported in Appendix 4.B together with the dominating measure. The normalization of the weights leads to a term $T^{-n} = \left(\sum S_h^{(a)} + \sum S_\ell^{(na)}\right)^{-n}$ in the expression of the joint density, which makes it impossible to factorize the density according to the allocated and non-allocated components. As in Beraha et al. (2022), to overcome this issue, we introduce an auxiliary random variable $u \,|\, T \sim \text{Gamma}(n, T)$. The joint density of data and parameters is then

$$\begin{aligned}
&p(\boldsymbol{y}, \boldsymbol{c}, \boldsymbol{\eta}, \boldsymbol{\mu}^{(a)}, \boldsymbol{\mu}^{(na)}, \boldsymbol{S}^{(a)}, \boldsymbol{S}^{(na)} \boldsymbol{\Delta}^{(a)}, \boldsymbol{\Delta}^{(na)}, \Sigma, \Lambda, \boldsymbol{\psi}, \boldsymbol{\phi}, \tau, u) = \\
&\quad \frac{u^{n-1}}{\Gamma(n)} \left[\prod_{i=1}^{n} \mathcal{N}_p(y_i \,|\, \Lambda\eta_i, \Sigma)\right] \left[\prod_{h=1}^{k} e^{-uS_h^{(a)}} (S_h^{(a)})^{n_h} \text{Ga}(S_h^{(a)} \,|\, \alpha, 1) \text{IW}(\Delta_h^{(a)} \,|\, \nu_0, \Psi_0)\right. \\
&\qquad \left. \times \prod_{i:c_i=h} \mathcal{N}_d(\eta_i \,|\, \mu_h^{(a)}, \Delta_h^{(a)})\right] \left[\prod_{h=1}^{\ell} e^{-uS_h^{(na)}} \text{Ga}(S_h^{(na)} \,|\, \alpha, 1) \text{IW}(\Delta_h^{(na)} \,|\, \nu_0, \Psi_0)\right] \\
&\quad f_{\text{DPP}}^{\text{app}}(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \,|\, \rho, \Lambda, K_0; R) \prod_{j=1}^{p} \left[\text{inv-Ga}(\sigma_j^2 \,|\, a_\sigma, b_\sigma) \prod_{h=1}^{d} \mathcal{N}(\lambda_{jh} \,|\, 0, \psi_{jh}\phi_{jh}^2\tau)\right. \\
&\qquad \left. \times \prod_{h=1}^{d} \text{Exp}(\Psi_{jd} \,|\, 1/2)\right] \text{Dir}(vec(\phi) \,|\, a) \text{Ga}(\tau \,|\, pda, 1/2)
\end{aligned}$$

Then, a Metropolis-within-Gibbs algorithm can be summarized in the following steps.

1. *Update of* (ψ, τ, ϕ). Following Bhattacharya et al. (2015) sample

$$\psi_{jh} \,|\, \lambda_{jh}, \phi_{jh}, \tau \overset{\text{ind}}{\sim} \text{giG}\left(\frac{1}{2}, 1, \frac{\lambda_{jh}^2}{\phi_{jh}^2 \tau^2}\right)$$

$$\tau \,|\, \phi, \Lambda \sim \text{giG}\left(p \cdot d \cdot (a-1), 1, 2 \sum_{\substack{j=1:p \\ h=1:d}} \frac{|\lambda_{jh}|}{\phi_{jh}}\right)$$

$$\phi_{jh} = \frac{T_{jh}}{T}, \quad T_{jh} \overset{\text{ind}}{\sim} \text{giG}(\, a-1, 1, 2|\lambda_{jh}|\,), \quad T := \sum_{j,h} T_{jh},$$

   for $j = 1, \dots, p$, $h = 1, \dots, d$, where giG denotes the generalized inverse-Gaussian distribution.

2. *Update of* Λ. Sample from the full conditional density

$$p(\Lambda \,|\, \cdots) \propto p(\boldsymbol{y} \,|\, \Lambda, \boldsymbol{\eta}, \Sigma) p(\Lambda \,|\, \phi, \tau, \psi) f_{\text{DPP}}^{\text{app}}(\boldsymbol{\mu} \,|\, \rho, \Lambda, K_0; R)$$

   using Metropolis-Hastings step, see Section 4.3.3 for further details.

3. *Update of* Σ. Sample each σ_j^2 independently from

$$\sigma_j^2 \mid y^{(j)}, \boldsymbol{\eta}, \lambda^{(j)} \overset{\text{ind}}{\sim} \text{inv}-\text{Ga}\left(\frac{n}{2}+a_\sigma, \frac{1}{2}\sum_{i=1}^{n}\left(y_i^{(j)} - \lambda^{(j)T}\eta_i\right)^2 + b_\sigma\right)$$

   where $\lambda^{(j)^\top}$ is the j-th row of Λ and $y^{(j)} = (y_{1j}, \dots, y_{nj})^\top$.

4. *Update of the non-allocated variables* $(\boldsymbol{\mu}^{(na)}, \boldsymbol{s}^{(na)}, \boldsymbol{\Delta}^{(na)})$. Following Beraha et al. (2022) we disintegrate the joint full conditional of the non-allocated variables as

$$p(\boldsymbol{\mu}^{(na)}, \boldsymbol{s}^{(na)}, \boldsymbol{\Delta}^{(na)} \mid \text{rest}) = p(\boldsymbol{\mu}^{(na)} \mid \text{rest})p(\boldsymbol{s}^{(na)} \mid \boldsymbol{\mu}^{(na)}, \text{rest})p(\boldsymbol{\Delta}^{(na)} \mid \boldsymbol{\mu}^{(na)}, \text{rest}),$$

   where "rest" identifies all the variables except for $(\boldsymbol{\mu}^{(na)}, \boldsymbol{s}^{(na)}, \boldsymbol{\Delta}^{(na)})$. Then $\boldsymbol{\mu}^{(na)} \mid \text{rest}$ is a Gibbs point process with density

$$p(\{\mu_1^{(na)}, \dots, \mu_\ell^{(na)}\} \mid \text{rest}) \propto f_{\text{DPP}}^{\text{app}}(\{\mu_1^{(na)}, \dots, \mu_\ell^{(na)}\} \cup \boldsymbol{\mu}^{(a)} \mid \rho, \Lambda, K_0; R)\psi(u)^\ell$$

   where $\psi(u) = \mathbb{E}[e^{-uS}]$. We employ the birth-death Metropolis-Hastings algorithm in Geyer and Møller (1994) to sample from this point process density. Given $\boldsymbol{\mu}^{(na)}$ it is straightforward to show

$$\begin{aligned}\Delta_1^{(na)}, \dots, \Delta_\ell^{(na)} \mid \cdots &\overset{\text{iid}}{\sim} \text{IW}_d(\Psi_0, \nu)\\ S_1^{(na)}, \dots, S_\ell^{(na)} \mid \cdots &\overset{\text{iid}}{\sim} \text{Gamma}(\alpha, 1+u)\end{aligned}$$

5. *Update of the allocated variables* $(\boldsymbol{\mu}^{(a)}, \boldsymbol{s}^{(a)}, \boldsymbol{\Delta}^{(a)})$. Let k be the number of unique values in $\boldsymbol{c}$ and assume that the active components are the first k. We can sample the allocated variables using a Gibbs scan. It is trivial to show that

$$\begin{aligned}S_h^{(a)} \mid \cdots &\overset{\text{ind}}{\sim} \text{Gamma}(\alpha + n_h, 1+u)\\ \Delta_h^{(a)} \mid \cdots &\overset{\text{ind}}{\sim} \text{IW}_d\left(\Psi_0 + \sum_{i:c_i=h}(\eta_i - \mu_h^{(a)})(\eta_i - \mu_h^{(a)})^\top, \nu + n_h\right).\end{aligned}$$

   The full conditional of $\boldsymbol{\mu}^{(a)}$ is proportional to

$$p(\boldsymbol{\mu}^{(a)} \mid \cdots) \propto f_{\text{DPP}}^{\text{app}}(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \mid \rho, \Lambda, K_0; R)\prod_{h=1}^{k}\prod_{i:c_i=h}\mathcal{N}_d(\eta_i \mid \mu_h^{(a)}, \Delta_h^{(a)})$$

   and we use a Metropolis-Hastings step to sample from it.

6. *Update of the latent allocation variables* $\boldsymbol{c}$. We found it useful to marginalize over the η_i's to get better mixing chains. Hence, we can sample each c_i independently from a discrete distribution over $\{1, \dots, k+\ell\}$ with weights ω_{ih}:

$$\begin{aligned}\omega_{ih} &\propto S_h^{(a)}\mathcal{N}_p(y_i \mid \Lambda\mu_h^{(a)}, \Sigma + \Lambda\Delta_h^{(a)}\Lambda^\top), && h = 1, \dots, k\\ \omega_{ik+h} &\propto S_h^{(na)}\mathcal{N}_p(y_i \mid \Lambda\mu_h^{(na)}, \Sigma + \Lambda\Delta_h^{(na)}\Lambda^\top), && h = 1, \dots, \ell\end{aligned}$$

   Each evaluation of the p-dimensional Gaussian density would require $\mathcal{O}(p^3)$ operations if some care is not taken. However, we take advantage from the special structure of the covariance matrix. Using Woodbury's matrix identity we have that

$$\left(\Sigma + \Lambda\Delta\Lambda^\top\right)^{-1} = \Sigma^{-1} - \Sigma^{-1}\Lambda\left(\Delta^{-1} + \Lambda^\top\Sigma^{-1}\Lambda\right)^{-1}\Lambda^\top\Sigma^{-1},$$

where the inversion is now required for a $d \times d$ matrix. Therefore, evaluating the quadratic form in the exponential can be done in a linear time with respect to p. Moreover, the determinant of the covariance matrix can be computed using the matrix determinant lemma as

$$\det(\Sigma + \Lambda\Delta\Lambda^\top) = \det(\Delta^{-1} + \Lambda^\top\Sigma^{-1}\Lambda)\det(\Delta)\det(\Sigma).$$

This is computed without additional cost by caching operations from the previous matrix inversion.

7. *Update of the ancillary variable* $u \sim \text{Gamma}(n, T)$

8. *Update of the latent scores* $\boldsymbol{\eta}$. For $i = 1, \dots, n$, sample each η_i independently from

$$\eta_i \mid \cdots \overset{\text{ind}}{\sim} \mathcal{N}_d(m_i\,,\, S_i)$$

where

$$S_i = \left(\Lambda^T\Sigma^{-1}\Lambda + (\Delta^{(a)}_{c_i})^{-1}\right)^{-1}, \quad m_i = S_i\left(\Lambda^T\Sigma^{-1}y_i + (\Delta^{(a)}_{c_i})^{-1}\mu^{(a)}_{c_i}\right)$$

4.3.3 Updating Λ using gradient-based MCMC algorithms

As mentioned in Section 4.2.1, sampling from Λ's full conditional is non-trivial. In particular, we found that random-walk Metropolis-Hastings led to very poor mixing of the MCMC chain, while the adaptive Metropolis-Hastings algorithm in Haario et al. (2001) is not feasible here due to the high dimensionality of Λ. Gradient-based MCMC algorithms such as the Metropolis adjusted Langevin Algorithm (Roberts and Tweedie, 1996) or Hamiltonian Monte Carlo (Neal et al., 2011) are thus the preferred solution here. The target full-conditional density is

$$\begin{aligned} p(\Lambda \mid \cdots) &\propto p(\boldsymbol{y} \mid \Lambda, \boldsymbol{\eta}, \Sigma) \\ &\quad \times |R|^{-n}\frac{e^{1-D^{\text{app}}}}{1-e^{-D^{\text{app}}}}\det[C^{\text{app}}](T\mu_1, \dots, T\mu_n) \\ &\quad \times p(\Lambda \mid \phi, \tau, \psi) \end{aligned} \tag{4.18}$$

Although not explicitly stated, D^{app} and C^{app} both depend on Λ. Automatic differentiation (AD, Griewank et al., 1989) provides an easy way to get gradients of functions. In a nutshell, AD exploits the chain rule of derivatives to obtain an analytically exact evaluation of the gradient of a function implemented in a software program. Therefore, we can get $\nabla\log(p(\Lambda \mid \cdots))$ simply by writing a function that evaluates $\log(p(\Lambda \mid \cdots))$. However, AD introduces a large number of additional parameters to the software execution to track all the computations involved. We found that in our particular case, computing $\nabla\log(p(\Lambda \mid \cdots))$ by means of AD is feasible only in trivial scenarios, i.e. up to $p = 50$ and $d = 3$ due to RAM memory requirements. See Figure 4.C.1. This is likely due to the large number of computations involving Λ needed to evaluate (4.18).

The next theorem provides the analytical expressions for $\nabla\log(p(\Lambda \mid \cdots))$ when the DPP involved is Gaussian-like and Whittle-Matern-like.

Theorem 4.2. *Under the Gaussian-like DPP prior, the gradient of the log-full conditional* *density of* Λ *equals*

$$\begin{aligned}\nabla \log p(\Lambda \mid \cdots) = \Sigma^{-1} \sum_{i=1}^{n} (y_i - \Lambda\eta_i)\eta_i^\top + \\ + (2\pi^2 c^{-\frac{2}{d}}) \sum_{k \in \mathbb{Z}_N^d} g^{(k)} \frac{\varphi(k)}{(1-\varphi(k))^2} \Big[\frac{1-\varphi(k)}{1-e^{-D^{\text{app}}}} - v_k^T \left(C^{\text{app}}\right)^{-1} u_k\Big] + \\ - \frac{1}{(\psi \odot \phi^2)\tau^2} \odot \Lambda\end{aligned}$$

where $\varphi(k)$ *refers to (4.10),* $\odot$ *denotes the elementwise (Hadamard) product,*

$$g^{(k)} = 2|\Lambda^T\Lambda|^{\frac{1}{d}} \Lambda(\Lambda^T\Lambda)^{-1}\Big[\frac{1}{d}k^T(\Lambda^T\Lambda)^{-1}k\mathbb{1}_d - k((\Lambda^T\Lambda)^{-1}k)^T\Big],$$

u_k *and* v_k *are* m*-dimensional column vectors for each* $k \in \mathbb{Z}^d$ *with entries*

$$(u_k)_j = e^{2\pi i k^T T\mu_j}, \quad (v_k)_j = e^{-2\pi i k^T T\mu_j}, \quad j = 1, \dots, m$$

and $C^{\text{app}} := C^{\text{app}}(T\mu_1, \dots, T\mu_n)$.

Similarly, under the Whittle-Matern-like DPP prior, the gradient of the log-full conditional density of Λ *equals*

$$\begin{aligned}\nabla \log p(\Lambda \mid \cdots) = \Sigma^{-1} \sum_{i=1}^{n} (y_i - \Lambda\eta_i)\eta_i^\top + \\ + 4\pi^2\alpha^2 \left(\nu + \frac{d}{2}\right) \sum_{k \in \mathbb{Z}_N^d} \frac{g^{(k)}}{a^{(k)}} \frac{\varphi(k)}{(1-\varphi(k))^2} \Big[\frac{1-\varphi(k)}{1-e^{-D^{\text{app}}}} - v_k^T \left(C^{\text{app}}\right)^{-1} u_k\Big] + \\ - \frac{1}{(\psi \odot \phi^2)\tau^2} \odot \Lambda\end{aligned}$$

where $\varphi(k)$ *refers to (4.11) and*

$$a^{(k)} = 1 + 4\pi^2\alpha^2|\Lambda^T\Lambda|^{\frac{1}{d}} k^T(\Lambda^T\Lambda)^{-1}k$$

Figure 4.C.1 in the Appendix reports a comparison of the memory requirements and the runtime execution for completing one iteration of our MCMC algorithm with $n = 100$ samples as p and d vary using the AD gradients or our analytical expressions. While memory requirement increases exponentially in d in both cases (this is to be expected given the sum over $\mathbb{Z}_N^d$ in the DPP density, see (4.16)), using the AD gradients requires roughly two orders of magnitude more memory, which makes a significant difference in practice. The runtimes are of one order of magnitude larger when using AD as well.

In our code, we use the MALA algorithm to update Λ, where the stepsize parameter is tuned running short preliminary chains to get an acceptance rate around 20%. In particular, we found that values between 10^{-8} and 10^{-10} usually give satisfactory results.

4.4 Simulation Studies

We illustrate the comparison between the APPLAM model with Gaussian-like DPP (4.10) and the Lamb model of Chandra et al. (2020) on two different sets of simulated datasets, referred to as simulation studies A and B. Let $t_p(m, \Sigma, \nu)$ denote the multivariate Student distribution with location m, scale matrix Σ and degrees of freedom ν, with density

$$f(x) = \frac{\Gamma((\nu+p)/2)}{\Gamma(\nu/2)\nu^{p/2}\pi^{p/2}|\Sigma|^{1/2}} \left[1 + \frac{1}{\nu}(x-m)^T\Sigma^{-1}(x-m)\right]^{-(\nu+p)/2}, \qquad x \in \mathbb{R}^p$$

In simulation study A, data are generated from the latent factor model (4.2) misspecified by replacing the Gaussian likelihood with the multivariate Student distribution $t_p(m, \Sigma, \nu)$, specifically

$$y_i \mid \eta_i, \Lambda, \Sigma \overset{\text{ind}}{\sim} t_p(y_i \mid \Lambda\eta_i,\, \Sigma,\, 3) \qquad i = 1, \dots, n$$

where $\Sigma = 0.5 \cdot I_p$ and Λ is fixed. Moreover, 50 samples of $\eta_i \in \mathbb{R}^d$ are drawn from each of $M = 4$ Gaussian kernels with means μ_h, $h = 1, \dots, M = 4$, and identity covariance matrices. Differently from scenario A, the simulation study B introduces the misspecification of the generating mechanism at the latent factors level rather than on the likelihood. Specifically, data are generated from the likelihood in (4.2), with $\Sigma = 0.5 \cdot I_p$ and Λ fixed. Moreover, 50 samples of $\eta_i \in \mathbb{R}^d$ are drawn from each of $M = 4$ multivariate Student kernels $t_d(\mu_h, I_d, 3)$, $h = 1, \dots, M = 4$. Moreover, for both the simulation studies A and B, we consider $p = \{100, 200, 400\}$ and $d = \{2, 5, 8\}$ and the data are standardized for the analysis.

4.4.1 Hyperparameters elicitation and MCMC details

The elicitation for the common hyperparameters in APPLAM and Lamb follows the default choices of Chandra et al. (2020). Specifically, we set $a_\sigma = 1, b_\sigma = 0.3$ in (4.14) and $a = 0.5$ in (4.15). Moreover, Lamb assumes a Dirichlet process location-scale mixture of Gaussian densities for the latent factors. In particular, the Normal inverse Wishart distribution is taken as base measure, with location μ_0, scale k, covariance matrix Ψ_0 and degrees of freedom ν_0. The default choice sets μ_0 as the null vector, $k = 0.001$, $\Psi_0 = \delta \cdot I_d$, with $\delta = 20$ and $\nu_0 = d+50$. Coherently, for APPLAM we set $\Psi_0 = \delta \cdot I_d$, $\delta = 20$ and $\nu_0 = d+50$ in (4.13). Finally, we set $\alpha = 1, \beta = 3$ in (4.17).

It is not straightforward to match the prior knowledge expressed by the concentration parameter of the Dirichlet process α_{DP} in Lamb and the pair (ρ, c) (see (4.10)), in APPLAM. Indeed, α_{DP} controls the prior distribution of the number of clusters (i.e., allocated components) under the Dirichlet Process prior, while the pair (ρ, c) controls the repulsiveness of the DPP and its intensity, that is the distribution of the number of components m in the mixture.

Therefore, we limit ourselves to evaluate empirically the robustness to the choice of these parameters in our simulations. In particular, we consider $\alpha_{DP} = \{0.1, 0.5, 1\}$, which correspond to an a priori expected number of clusters of $1.57, 3.63$, and 5.88 respectively. For our model instead, we set c_ρ such that $\rho = \rho_{max}/2$ as in Bianchini et al. (2020) and Beraha et al. (2022), and consider $\rho = \{5, 10, 20\}$. In particular, ρ equals to the expected number of components a priori.

For each run of the APPLAM model, we perform $2 \cdot 10^3$ burn-in iterations and $4 \cdot 10^3$ iterations with thinning equal to 5. For each run of the Lamb model, we perform 10^6 burn-in iterations and $5 \cdot 10^4$ iterations with thinning equal to 10. The poor mixing of the Lamb algorithm in our simulations (see Figure 4.C.2) demands a very long burn-in phase.

4.4.2 Comparison between the two models

Figures 4.4.1 and 4.4.2 show the posterior distribution of the number of clusters for our model and for LAMB in the two simulation settings. Moreover, in the Appendix, we report posterior summary statistics of the clustering such as the mode and mean of the number of clusters, the credible interval for the adjusted rand index (ARI) between the estimated and true clustering, and the ARI between the best partition (obtained by minimizing the Binder loss function) and the true one. See Tables 4.C.1 -4.C.3, and Tables 4.C.4) - 4.C.6 for simulations A and B, respectively.

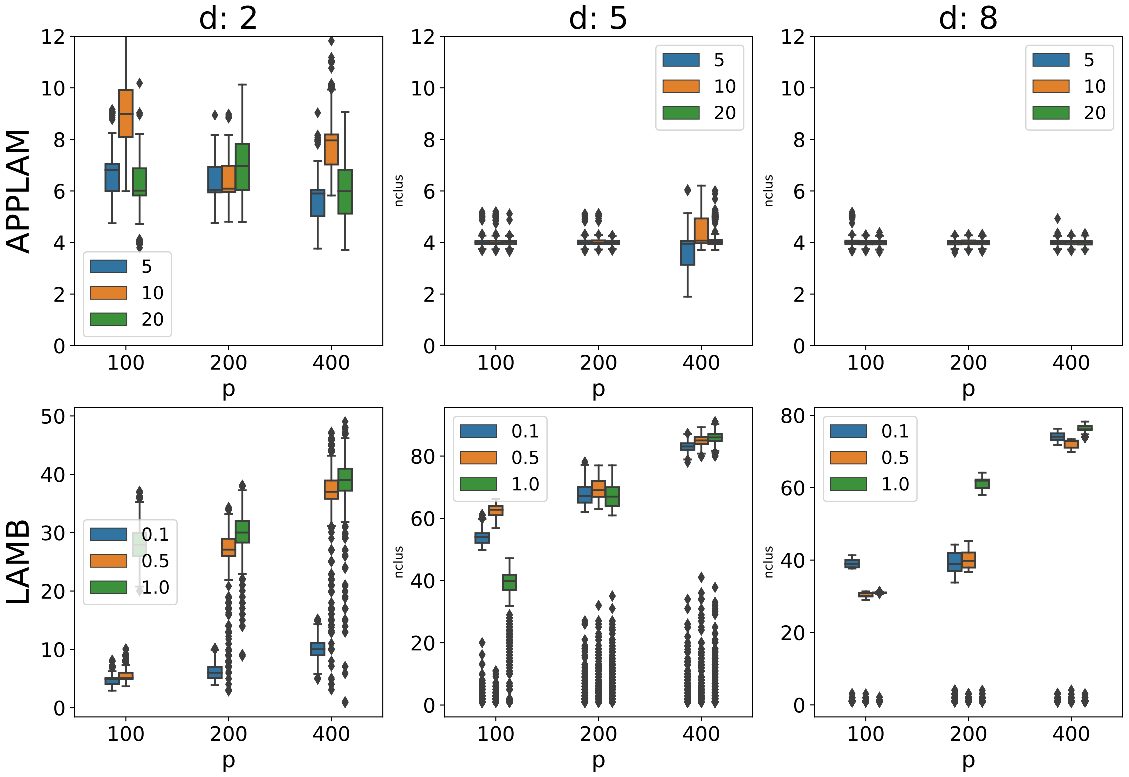


Figure 4.4.1: Posterior distribution of the number of clusters in simulation A, for APPLAM (top row) and LAMB (bottom row). Each panel corresponds to a different value of d and reports the posterior number of clusters as the dimension p and the model parameters vary.

For both data generation processes, all the analyzed datasets (that differ for the values of p, d) agree on the fact that Lamb is sensitive to α_{DP}. In general, we can see that Lamb tends to estimate a large number of clusters. On the other hand, APPLAM appears rather robust to the choice of ρ: not only the detected number of clusters is always reasonable (hardly more than 9, but most of the time the MCMC finds the correct number of clusters.

When looking at the ARI between the best clustering and the true one, as well as the credible intervals of the ARI of the clustering, we see that for 24 out of 27 datasets simulated according to scenario A, APPLAM provides a better clustering than Lamb. In particular, in several settings the ARI between the best clustering and the true one for APPLAM is equal or close to 1.0 (i.e., perfect cluster detection) while for LAMB it is smaller than 0.1 (i.e., almost random guessing). For scenario B instead, APPLAM provides a better clustering than Lamb 17 times out of 27.

4.5 Discussion

Motivated by the problem of clustering high-dimensional data, in this chapter we introduced a new class of anisotropic determinantal point processes, which induce repulsiveness within the Λ-weighted Euclidean metric $\|x-y\|_{\Lambda^\top \Lambda} = (x-y)^\top \Lambda^\top \Lambda (x-y)$. In particular, we extend the general construction in Lavancier et al. (2015) and obtain easy-to-check conditions that guarantee the existence of the DPP, the existence of a density with respect to the unit rate Poisson process and its spectral density.

In our clustering framework, the DPP is assumed as prior for latent d-dimensional pa-

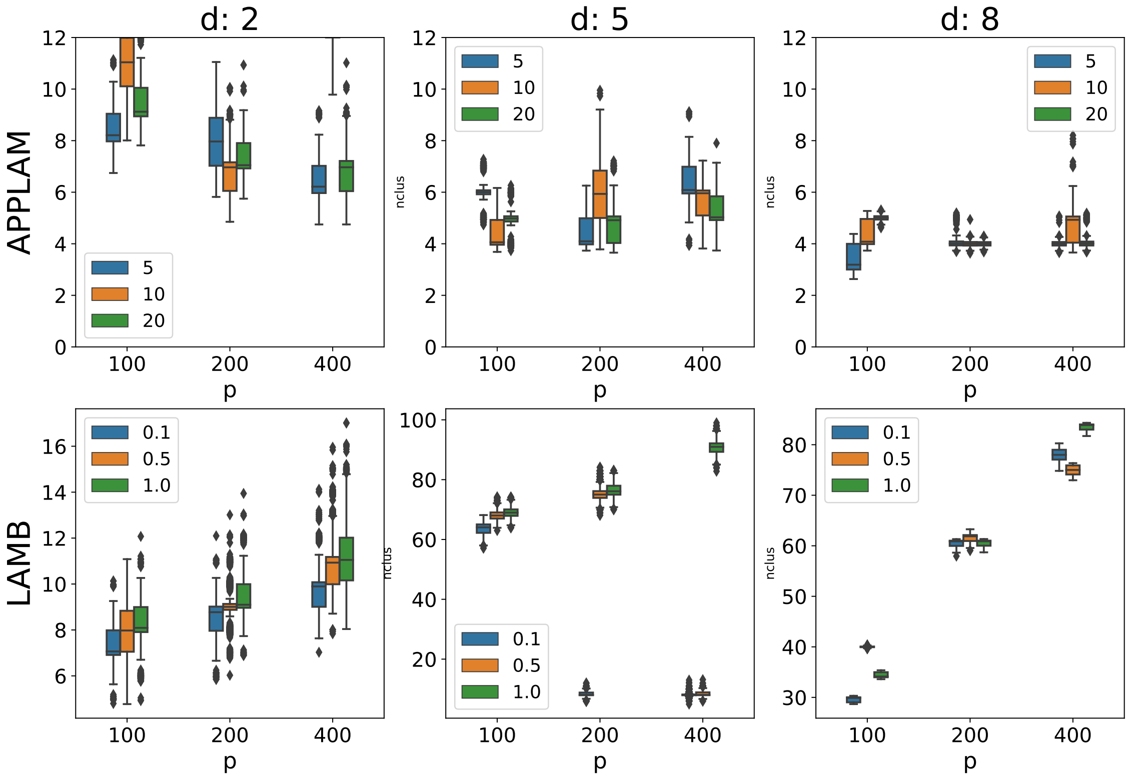


Figure 4.4.2: Posterior distribution of the number of clusters in simulation B, for APPLAM (top row) and LAMB (bottom row). Each panel corresponds to a different value of d and reports the posterior number of clusters as the dimension p and the model parameters vary.

rameters $\mu_1, \dots, \mu_m$ whereas $p \gg d$ dimensional observations are associated with cluster centers $\Lambda\mu_1, \dots, \Lambda\mu_m$. Therefore, repelling the μ_h' s with the Λ-weighted metric ensures that our prior forces well-separated clusters at the observational level. Throughout several simulations, we empirically validate our model and show that assuming a repulsive prior yields more robust clustering and overall better performance when the model is misspecified.

Several questions are open for future discussion. First, the approximation of the DPP density has a cost that scales exponentially in the latent dimension d. While for several applications where p is moderate (i.e., less than $1,000$) one would expect that a sufficiently small d (that is, $d \leq 10$) provides reasonable results, when p is extremely large, such as in the case of single-cell data, it is likely that a larger latent dimension d would be needed. In this case, other approximations of the DPP density not based on the spectral representation might be worth considering, such as the one in Poinas and Lavancier (2021) or the one in Bardenet and Titsias (2015). Second, the MCMC move that updates the non-allocated components is based on a birth-death move which either adds or deletes one point of $\mu_1, \dots, \mu_m$ at every iteration. Moreover, the new point is sampled from a uniform distribution over the support of the DPP. We might want to consider alternatives to this update, where one can add or remove multiple points and where the new points are sampled by taking into account the datapoints as well. Finally, it would also be interesting to consider more general (repulsive) models to tackle feature sampling and trait allocation models.

Appendix

4.A Proofs

4.A.1 Proof of Theorem 4.1

Proof. Since Λ is full rank, the conditional distribution (4.6) for the random variable Y is well-posed. In fact, $\Lambda^T\Lambda$ is positive definite, thus it is invertible. Denoting with $|\Lambda^T\Lambda|$ the determinant of $\Lambda^T\Lambda$, we explicitly compute the density $h(x)$ from model (4.6). We have:

$$h(x) = \int_0^\infty p(x\,|\,w)\cdot p(w)dw = \int_0^\infty \frac{|\Lambda^T\Lambda|^{\frac{1}{2}}}{(2\pi)^{\frac{d}{2}} w^{\frac{d}{2}}} \exp\Bigg(-\frac{x^T\Lambda^T\Lambda x}{2w}\Bigg)\cdot p(w)dw$$

Therefore, we derive

$$h(x) = \frac{|\Lambda^T\Lambda|^{\frac{1}{2}}}{(2\pi)^{\frac{d}{2}}}\,\mathbb{E}\Bigg[W^{-\frac{d}{2}}\exp\Bigg(-\frac{||\Lambda x||^2}{2W}\Bigg)\Bigg], \qquad x\in\mathbb{R}^d \tag{4.19}$$

Consequently, since $K_0(x) = \rho h(x)/h(0)$, we have

$$K_0(x) = \frac{\rho}{\mathbb{E}\big[W^{-\frac{d}{2}}\big]}\,\mathbb{E}\Bigg[W^{-\frac{d}{2}}\exp\Bigg(-\frac{||\Lambda x||^2}{2W}\Bigg)\Bigg], \qquad x\in\mathbb{R}^d$$

and, since $\varphi = \mathcal{F}(K_0)$, we compute

$$\begin{aligned}
\varphi(x) &= \int_{\mathbb{R}^d} e^{-2\pi i x^T y}\,K_0(y)dy = \\
&= \frac{\rho}{h(0)}\int_{\mathbb{R}^d} e^{-2\pi i x^T y}\,\frac{|\Lambda^T\Lambda|^{\frac{1}{2}}}{(2\pi)^{\frac{d}{2}}}\,\mathbb{E}\Bigg[W^{-\frac{d}{2}}\exp\Bigg(-\frac{||\Lambda y||^2}{2W}\Bigg)\Bigg]dy = \\
&= \frac{\rho\,|\Lambda^T\Lambda|^{\frac{1}{2}}}{h(0)\,(2\pi)^{\frac{d}{2}}}\int_{\mathbb{R}^d} e^{-2\pi i x^T y}\int_0^\infty w^{-\frac{d}{2}}\exp\Bigg(-\frac{||\Lambda y||^2}{2w}\Bigg)p(w)dw\,dy = \\
&= \frac{\rho\,|\Lambda^T\Lambda|^{\frac{1}{2}}}{h(0)\,(2\pi)^{\frac{d}{2}}}\int_0^\infty\int_{\mathbb{R}^d} w^{-\frac{d}{2}}\exp\Bigg(-2\pi i x^T y - \frac{y^T\Lambda^T\Lambda y}{2w}\Bigg)dy\,p(w)dw = \\
&= \frac{\rho\,|\Lambda^T\Lambda|^{\frac{1}{2}}}{h(0)\,(2\pi)^{\frac{d}{2}}}\int_0^\infty\int_{\mathbb{R}^d} w^{-\frac{d}{2}}\exp\Bigg(-\frac{1}{2}\Bigg[y^T\frac{\Lambda^T\Lambda}{w}y + 4\pi i x^T y\Bigg]\Bigg)dy\,p(w)dw
\end{aligned}$$

The term in brackets can be handled as follows

$$\begin{aligned}
[...] =& \big(y-(-2\pi i w(\Lambda^T\Lambda)^{-1}x)\big)^T\,\frac{\Lambda^T\Lambda}{w}\,\big(y-(-2\pi i w(\Lambda^T\Lambda)^{-1}x)\big) \\
&+ 4\pi^2 x^T w(\Lambda^T\Lambda)^{-1}x
\end{aligned}$$

Plugging in the previous computation, it results in

$$\varphi(x) = \frac{\rho}{h(0)} \int_0^{\infty} \exp(-2\pi^2 w x^T (\Lambda^T \Lambda)^{-1} x)\, p(w) dw$$

Summing up, we derive

$$\varphi(x) = \frac{\rho}{h(0)}\, \mathbb{E}\Big[\exp\big(-2\pi^2 W x^T (\Lambda^T \Lambda)^{-1} x\big)\Big], \qquad x \in \mathbb{R}^d$$

Observe that, since $h(x)$ is the density of a real random variable and using Fourier transform properties, then

$$K_0(x) = \rho\, \frac{h(x)}{h(0)} \in L^1(\mathbb{R}^d), \qquad \varphi = \mathcal{F}(K_0) \in L^1(\mathbb{R}^d)$$

Therefore, to guarantee the existence of the anisotropic determinantal point process with kernel K_0 and spectral density $\varphi = \mathcal{F}(K_0)$, we refer to Corollary (3.3) of Lavancier et al. (2015): since $K_0 \in L^1(\mathbb{R}^d), \varphi = \mathcal{F}(K_0)$ and $\varphi \in L^1(\mathbb{R}^d)$, we just need to ensure

$$\varphi(x) \leq 1, \qquad \forall x \in \mathbb{R}^d$$

So, we need to require

$$\max \varphi(x) = \varphi(0) = \frac{\rho}{h(0)} \leq 1$$

Finally, the existence condition expresses as

$$\rho \leq \rho_{max}$$

with

$$\rho_{max} = h(0) = \frac{|\Lambda^T \Lambda|^{\frac{1}{2}}}{(2\pi)^{\frac{d}{2}}}\, \mathbb{E}\Big[W^{-\frac{d}{2}}\Big]$$

□

4.A.2 Proof of Corollary 4.1

Proof. Let $c > 0$. In Theorem 4.1, set

$$W = |\Lambda^T \Lambda|^{\frac{1}{d}} \cdot c^{-\frac{2}{d}}$$

Consequently, $W^{-\frac{d}{2}} = |\Lambda^T \Lambda|^{-\frac{1}{2}} \cdot c$, and from Equation (4.19), we derive

$$h(x) = \frac{c}{(2\pi)^{\frac{d}{2}}} \cdot \exp\bigg(-\frac{||\Lambda x||^2}{2|\Lambda^T \Lambda|^{\frac{1}{d}}\, c^{-\frac{2}{d}}}\bigg), \qquad x \in \mathbb{R}^d$$

From Equations (4.8) and (4.9),

$$K_0(x) = \rho \cdot \exp\bigg(-\frac{||\Lambda x||^2}{2|\Lambda^T \Lambda|^{\frac{1}{d}}\, c^{-\frac{2}{d}}}\bigg), \qquad x \in \mathbb{R}^d$$

$$\varphi(x) = \rho\, \frac{(2\pi)^{\frac{d}{2}}}{c} \cdot \exp\Big(-2\pi^2 |\Lambda^T \Lambda|^{\frac{1}{d}} c^{-\frac{2}{d}} x^T (\Lambda^T \Lambda)^{-1} x\Big), \qquad x \in \mathbb{R}^d$$

From (4.7), the existence condition requires $\rho \leq \rho_{max}$, with

$$\rho_{max} = \frac{c}{(2\pi)^{d/2}}$$

□

4.A.3 Proof of Corollary 4.2

Proof. Let $\nu > 0$ and $\alpha > 0$. In Theorem 4.1, set

$$W \sim \text{Gamma}\left(\nu + \frac{d}{2}, \frac{1}{2|\Lambda^T\Lambda|^{\frac{1}{d}}\alpha^2}\right)$$

Applying Equation (4.19), we explicitly compute $h(x)$ as

$$h(x) = \frac{|\Lambda^T\Lambda|^{\frac{1}{2}}}{(2\pi)^{\frac{d}{2}}} \int_0^\infty \left[w^{-\frac{d}{2}} \exp\left(-\frac{||\Lambda x||^2}{2w}\right) \left(\frac{1}{2|\Lambda^T\Lambda|^{\frac{1}{d}}\alpha^2}\right)^{\nu+\frac{d}{2}} \frac{w^{\nu+\frac{d}{2}-1}}{\Gamma\big(\nu+\frac{d}{2}\big)} \cdot \right.$$
$$\left. \cdot \exp\left(-\frac{w}{2|\Lambda^T\Lambda|^{\frac{1}{d}}\alpha^2}\right)\right] dw =$$
$$= \frac{|\Lambda^T\Lambda|^{-\frac{\nu}{d}}}{(2\pi)^{\frac{d}{2}}(2\alpha^2)^{\nu+\frac{d}{2}}\Gamma\big(\nu+\frac{d}{2}\big)} \int_0^\infty w^{\nu-1} \exp\left[-\frac{1}{2}\left(\frac{1}{|\Lambda^T\Lambda|^{\frac{1}{d}}\alpha^2} w + \frac{||\Lambda x||^2}{w}\right)\right] dw$$

We need to recall the generalized inverse Gaussian distribution (GIG) on $[0,\infty)$ and its common parametrization. Let $p \in \mathbb{R}$, $a > 0$ and $b > 0$; then the generalized inverse Gaussian distribution with parameters (p, a, b) has probability density function

$$f(x) = \frac{(a/b)^{\frac{p}{2}}}{2K_p(\sqrt{ab})} x^{p-1} \exp\left(-\frac{1}{2}\left[ax + \frac{b}{x}\right]\right),\ x > 0 \tag{4.20}$$

where K_p is the modified Bessel function of the second kind. We indicate such a distribution with $\text{giG}(p, a, b)$.

Resuming the computation of $h(x)$, we can identify the density of a generalized inverse Gaussian distribution with parameters

$$p = \nu, \qquad a = \frac{1}{|\Lambda^T\Lambda|^{\frac{1}{d}}\alpha^2}, \qquad b = ||\Lambda x||^2$$

Therefore, carrying on the computation

$$h(x) = \frac{|\Lambda^T\Lambda|^{-\frac{\nu}{d}}}{(2\pi)^{\frac{d}{2}}(2\alpha^2)^{\nu+\frac{d}{2}}\,\Gamma\big(\nu+\frac{d}{2}\big)} \cdot \left(|\Lambda^T\Lambda|^{\frac{1}{d}}\alpha^2\,||\Lambda x||^2\right)^{\frac{\nu}{2}} \cdot 2\,K_\nu\left(\frac{||\Lambda x||}{|\Lambda^T\Lambda|^{\frac{1}{2d}}\,\alpha}\right)$$

Summing up the result of the computation

$$h(x) = \frac{1}{(\sqrt{\pi}\,\alpha)^d\ 2^{\nu+d-1}\ \Gamma\big(\nu+\frac{d}{2}\big)} \cdot \left|\left|\frac{\Lambda x}{\alpha\,|\Lambda^T\Lambda|^{\frac{1}{2d}}}\right|\right|^{\nu} \cdot K_\nu\left(\left|\left|\frac{\Lambda x}{\alpha\,|\Lambda^T\Lambda|^{\frac{1}{2d}}}\right|\right|\right)$$

Note that, as $x \to 0$, then $x^\nu K_\nu(x) \to 2^{\nu-1}\Gamma(\nu)$. Consequently, the kernel K_0 is

$$K_0(x) = \rho\,\frac{2^{1-\nu}}{\Gamma(\nu)} \cdot \left|\left|\frac{\Lambda x}{\alpha\,|\Lambda^T\Lambda|^{\frac{1}{2d}}}\right|\right|^{\nu} \cdot K_\nu\left(\left|\left|\frac{\Lambda x}{\alpha\,|\Lambda^T\Lambda|^{\frac{1}{2d}}}\right|\right|\right)$$

To derive the spectral density $\varphi = \mathcal{F}(K_0)$, some additional computations are needed starting from Equation (4.9):

$$
\begin{aligned}
\varphi(x) &= \frac{\rho}{h(0)} \int_0^{\infty} \exp\bigl(-2\pi^2 w x^T (\Lambda^T\Lambda)^{-1} x\bigr) \left(\frac{1}{2|\Lambda^T\Lambda|^{\frac{1}{d}}\alpha^2}\right)^{\nu+\frac{d}{2}} \frac{w^{\nu+\frac{d}{2}-1}}{\Gamma\bigl(\nu+\frac{d}{2}\bigr)} \exp\left(-\frac{w}{2|\Lambda^T\Lambda|^{\frac{1}{d}}\alpha^2}\right) dw \\
&= \frac{\rho}{h(0)\bigl(2|\Lambda^T\Lambda|^{\frac{1}{d}}\alpha^2\bigr)^{\nu+\frac{d}{2}}} \int_0^{\infty} \frac{w^{\nu+\frac{d}{2}-1}}{\Gamma\bigl(\nu+\frac{d}{2}\bigr)} \exp\left[-\left(2\pi^2 x^T(\Lambda^T\Lambda)^{-1}x + \frac{1}{2|\Lambda^T\Lambda|^{\frac{1}{d}}\alpha^2}\right) w\right] dw \\
&= \frac{\rho}{h(0)\bigl(2|\Lambda^T\Lambda|^{\frac{1}{d}}\alpha^2\bigr)^{\nu+\frac{d}{2}}} \cdot \frac{1}{\left(2\pi^2 x^T(\Lambda^T\Lambda)^{-1}x + \frac{1}{2|\Lambda^T\Lambda|^{1/d}\alpha^2}\right)^{\nu+\frac{d}{2}}} \\
&= \frac{\rho}{h(0)} \cdot \frac{1}{\bigl(1+4\pi^2\alpha^2|\Lambda^T\Lambda|^{\frac{1}{d}}\, x^T(\Lambda^T\Lambda)^{-1}x\bigr)^{\nu+\frac{d}{2}}}
\end{aligned}
$$

Finally, the spectral density $\varphi = \mathcal{F}(K_0)$ is

$$
\varphi(x) = \rho\,\frac{\Gamma\bigl(\nu+\frac{d}{2}\bigr)}{\Gamma(\nu)}\,\frac{(2\sqrt{\pi}\alpha)^d}{\bigl(1+4\pi^2\alpha^2|\Lambda^T\Lambda|^{\frac{1}{d}}\, x^T(\Lambda^T\Lambda)^{-1}x\bigr)^{\nu+\frac{d}{2}}}
$$

To sum up, we report the kernel K_0 and the spectral density φ of this model

$$
K_0(x) = \rho\,\frac{2^{1-\nu}}{\Gamma(\nu)} \cdot \left\|\frac{\Lambda x}{\alpha\,|\Lambda^T\Lambda|^{\frac{1}{2d}}}\right\|^{\nu} \cdot K_{\nu}\left(\left\|\frac{\Lambda x}{\alpha\,|\Lambda^T\Lambda|^{\frac{1}{2d}}}\right\|\right)
$$

$$
\varphi(x) = \rho\,\frac{\Gamma\bigl(\nu+\frac{d}{2}\bigr)}{\Gamma(\nu)}\,\frac{(2\sqrt{\pi}\alpha)^d}{\bigl(1+4\pi^2\alpha^2|\Lambda^T\Lambda|^{\frac{1}{d}}\, x^T(\Lambda^T\Lambda)^{-1}x\bigr)^{\nu+\frac{d}{2}}}
$$

The existence condition, from Equation (4.7), requires $\rho \le \rho_{max}$, with

$$
\rho_{max} = \frac{\Gamma(\nu)}{\Gamma\bigl(\nu+\frac{d}{2}\bigr)\,(2\sqrt{\pi}\alpha)^d}
$$

□

4.A.4 Proof of Theorem 4.2

Proof. Consider

$$
\log p(\Lambda \mid \cdots) \propto \log p(\boldsymbol{y} \mid \Lambda, \boldsymbol{\eta}, \Sigma) + \log f_{\text{DPP}}^{\text{app}}(\boldsymbol{\mu} \mid \Lambda) + \log p(\Lambda \mid \phi, \tau, \psi) \tag{4.21}
$$

The only term depending on the selected anisotropic DPP is the second one. Since it is the most complex term, we derive it using the two lemmas below. For the first term of (4.21),

$$
\begin{aligned}
\nabla \log p(\boldsymbol{y} \mid \Lambda, \boldsymbol{\eta}, \Sigma) &= \nabla\left(-\frac{1}{2}\sum_{i=1}^{n}(y_i - \Lambda\eta_i)^{\top}\Sigma^{-1}(y_i - \Lambda\eta_i)\right) \\
&= \Sigma^{-1} \cdot \sum_{i=1}^{n}(y_i - \Lambda\eta_i)\eta_i^T
\end{aligned}
$$

While for the last term

$$\nabla \log p(\Lambda \mid \phi, \tau, \psi) = -\frac{1}{(\psi \odot \phi^2)\tau^2} \odot \Lambda$$

For the second term of (4.21),

$$\begin{aligned}\nabla \log f_{\mathrm{DPP}}^{\mathrm{app}}(\boldsymbol{\mu} \mid \Lambda) &= \nabla\big[-D^{\mathrm{app}} - \log(1-e^{-D^{\mathrm{app}}}) + \log\det[C^{\mathrm{app}}](T\mu_1,\dots,T\mu_n)\big] \\ &= -\nabla D^{\mathrm{app}} - \nabla\log(1-e^{-D^{\mathrm{app}}}) + \nabla\log\det[C^{\mathrm{app}}](T\mu_1,\dots,T\mu_n) \\ &= -\frac{1}{1-e^{-D^{\mathrm{app}}}}\nabla D^{\mathrm{app}} + \nabla\log\det[C^{\mathrm{app}}](T\mu_1,\dots,T\mu_n) \qquad (4.22)\end{aligned}$$

To handle equation (4.22), the terms ∇D^{app} and $\nabla\log\det[C^{\mathrm{app}}](T\mu_1,\dots,T\mu_n)$ are to be computed.

Lemma 4.1. *For the Gaussian-like DPP prior,*

$$\nabla D^{\mathrm{app}} = \sum_{k\in\mathbb{Z}_N^d} \frac{\varphi(k)}{1-\varphi(k)}(-2\pi^2 c^{-\frac{2}{d}})g^{(k)} \qquad (4.23)$$

where $\varphi(k)$ refers to (4.10). For the Whittle-Matern-like DPP prior,

$$\nabla D^{\mathrm{app}} = \sum_{k\in\mathbb{Z}_N^d} \frac{\varphi(k)}{1-\varphi(k)}4\pi^2\alpha^2\left(-\nu-\frac{d}{2}\right)\frac{g^{(k)}}{a^{(k)}} \qquad (4.24)$$

where $\varphi(k)$ refers to (4.11).

Proof. Write

$$\nabla D^{\mathrm{app}} = -\sum_{k\in\mathbb{Z}_N^d}\nabla\log(1-\varphi(k)) = \sum_{k\in\mathbb{Z}_N^d}\frac{1}{1-\varphi(k)}\nabla\varphi(k).$$

For the Gaussian-like DPP prior, from (4.10), observe that, for $k\in\mathbb{Z}^d$

$$\begin{aligned}\nabla\varphi(k) &= \frac{\rho}{c}(2\pi)^{\frac{d}{2}}\nabla\exp\left(-2\pi^2|\Lambda^T\Lambda|^{\frac{1}{d}}c^{-\frac{2}{d}}k^T(\Lambda^T\Lambda)^{-1}k\right) \\ &= \varphi(k)\left(-2\pi^2c^{-\frac{2}{d}}\right)\nabla\left(|\Lambda^T\Lambda|^{\frac{1}{d}}k^T(\Lambda^T\Lambda)^{-1}k\right)\end{aligned}$$

For the Whittle-Matern-like DPP prior, from (4.11), observe that, for $k\in\mathbb{Z}^d$

$$\begin{aligned}\nabla\varphi(k) &= \rho\frac{\Gamma(\nu+\frac{d}{2})}{\Gamma(\nu)}(2\sqrt{\pi}\alpha)^d\nabla\left(\left(1+4\pi^2\alpha^2|\Lambda^T\Lambda|^{\frac{1}{d}}\,k^T\left(\Lambda^T\Lambda\right)^{-1}k\right)^{-\nu-\frac{d}{2}}\right) \\ &= \frac{\varphi(k)}{a^{(k)}}4\pi^2\alpha^2\left(-\nu-\frac{d}{2}\right)\nabla\left(|\Lambda^T\Lambda|^{\frac{1}{d}}k^T(\Lambda^T\Lambda)^{-1}k\right)\end{aligned}$$

Note that

$$\begin{aligned}\nabla\left(|\Lambda^T\Lambda|^{\frac{1}{d}}k^T(\Lambda^T\Lambda)^{-1}k\right) =&\nabla\left(|\Lambda^T\Lambda|^{\frac{1}{d}}\right)k^T(\Lambda^T\Lambda)^{-1}k+ \\ &+|\Lambda^T\Lambda|^{\frac{1}{d}}\nabla\left(k^T(\Lambda^T\Lambda)^{-1}k\right) = \\ =&\frac{1}{d}|\Lambda^T\Lambda|^{\frac{1}{d}-1}2|\Lambda^T\Lambda|\Lambda(\Lambda^T\Lambda)^{-1}k^T(\Lambda^T\Lambda)^{-1}k+ \\ &+|\Lambda^T\Lambda|^{\frac{1}{d}}\nabla\left(k^T(\Lambda^T\Lambda)^{-1}k\right)\end{aligned}$$

where, in the last step, formula (53) of Petersen and Pedersen (2012) is used. Then,

$$\begin{aligned}\nabla\left(k^T(\Lambda^T\Lambda)^{-1}k\right) &= \nabla\mathrm{tr}\left(k^T(\Lambda^T\Lambda)^{-1}k\right) = \\ &= \nabla\mathrm{tr}\left((\Lambda^T\Lambda)^{-1}kk^T\right) = -\Lambda(\Lambda^T\Lambda)^{-1}(2kk^T)(\Lambda^T\Lambda)^{-1}\end{aligned}$$

where, in the last step, formula (125) of Petersen and Pedersen (2012) is used. For the Gaussian-like DPP prior, this leads to

$$\nabla\varphi(k) = \varphi(k)(-4\pi^2c^{-\frac{2}{d}})|\Lambda^T\Lambda|^{\frac{1}{d}}\Lambda(\Lambda^T\Lambda)^{-1}\left[\frac{1}{d}k^T(\Lambda^T\Lambda)^{-1}k\mathbb{1}_d - kk^T(\Lambda^T\Lambda)^{-1}\right] \tag{4.25}$$

where $\mathbb{1}_d$ is $d\times d$ matrix of 1's. For the Whittle-Matern-like DPP prior, it follows

$$\nabla\varphi(k) = \frac{\varphi(k)}{a^{(k)}}8\pi^2\alpha^2\left(-\nu-\frac{d}{2}\right)|\Lambda^T\Lambda|^{\frac{1}{d}}\Lambda(\Lambda^T\Lambda)^{-1}\left[\frac{1}{d}k^T(\Lambda^T\Lambda)^{-1}k\mathbb{1}_d - kk^T(\Lambda^T\Lambda)^{-1}\right]$$

This concludes the proof. □

Lemma 4.2. *For the Gaussian-like DPP prior,*

$$\nabla\log\det[C^{\mathrm{app}}] = (-2\pi^2c^{-\frac{2}{d}})\sum_{k\in\mathbb{Z}_N^d}\frac{\varphi(k)}{(1-\varphi(k))^2}g^{(k)}v_k^T(C^{\mathrm{app}})^{-1}u_k \tag{4.26}$$

where $\varphi(k)$ refers to (4.10). For the Whittle-Matern-like DPP prior,

$$\nabla\log\det[C^{\mathrm{app}}] = 4\pi^2\alpha^2\left(-\nu-\frac{d}{2}\right)\sum_{k\in\mathbb{Z}_N^d}\frac{\varphi(k)}{(1-\varphi(k))^2}\frac{g^{(k)}}{a^{(k)}}v_k^T(C^{\mathrm{app}})^{-1}u_k \tag{4.27}$$

where $\varphi(k)$ refers to (4.11).

Proof.

$$\frac{\partial}{\partial\Lambda_{ij}}\log\det[C^{\mathrm{app}}] = \mathrm{tr}\left(\left(\frac{\partial}{\partial U}\log\det U\right)\Big|_{U=C^{\mathrm{app}}}\cdot\frac{\partial}{\partial\Lambda_{ij}}C^{\mathrm{app}}\right) \tag{4.28}$$

$$= \mathrm{tr}\left(\frac{1}{\det[C^{\mathrm{app}}]}\left(\frac{\partial}{\partial U}\det U\right)\Big|_{U=C^{\mathrm{app}}}\cdot\frac{\partial}{\partial\Lambda_{ij}}C^{\mathrm{app}}\right)$$

$$= \mathrm{tr}\left((C^{\mathrm{app}})^{-1}\cdot\frac{\partial}{\partial\Lambda_{ij}}C^{\mathrm{app}}\right) \tag{4.29}$$

where formula (137) of Petersen and Pedersen (2012) is used to get equation (4.28) and formula (49) of Petersen and Pedersen (2012) for equation (4.29). Now, note that

$$C^{\mathrm{app}} = \sum_{k\in\mathbb{Z}_N^d}\frac{\varphi(k)}{1-\varphi(k)}u_kv_k^T \tag{4.30}$$

where u_k, v_k are defined in the statement. It follows that, for the Gaussian-like DPP prior,

$$\frac{\partial}{\partial\Lambda_{ij}}C^{\mathrm{app}} = \sum_{k\in\mathbb{Z}_N^d}\frac{\varphi(k)}{(1-\varphi(k))^2}(-2\pi^2c^{-\frac{2}{d}})g_{ij}^{(k)}u_kv_k^T$$

where $g_{ij}^{(k)} = (g^{(k)})_{ij}$, while for the Whittle-Matern-like DPP prior,

$$\frac{\partial}{\partial\Lambda_{ij}}C^{\mathrm{app}} = \sum_{k\in\mathbb{Z}_N^d}\frac{\varphi(k)}{(1-\varphi(k))^2}4\pi^2\alpha^2\left(-\nu-\frac{d}{2}\right)\frac{g_{ij}^{(k)}}{a^{(k)}}u_kv_k^T$$

Then, back to equation (4.29), for the Gaussian-like DPP prior,

$$\begin{aligned}\frac{\partial}{\partial \Lambda_{ij}} \log\det[C^{\text{app}}] &= \operatorname{tr}\left((C^{\text{app}})^{-1} \cdot \frac{\partial}{\partial \Lambda_{ij}} C^{\text{app}}\right) \\ &= \operatorname{tr}\left(\sum_{k\in\mathbb{Z}_N^d} \frac{\varphi(k)}{(1-\varphi(k))^2}(-2\pi^2 c^{-\frac{2}{d}}) g_{ij}^{(k)} (C^{\text{app}})^{-1} u_k v_k^T\right) \\ &= \sum_{k\in\mathbb{Z}_N^d} \frac{\varphi(k)}{(1-\varphi(k))^2}(-2\pi^2 c^{-\frac{2}{d}}) g_{ij}^{(k)} \operatorname{tr}\left((C^{\text{app}})^{-1} u_k v_k^T\right) \\ &= (-2\pi^2 c^{-\frac{2}{d}}) \sum_{k\in\mathbb{Z}_N^d} \frac{\varphi(k)}{(1-\varphi(k))^2} g_{ij}^{(k)} v_k^T (C^{\text{app}})^{-1} u_k\end{aligned}$$

Similarly, for the Whittle-Matern-like DPP prior,

$$\frac{\partial}{\partial \Lambda_{ij}} \log\det[C^{\text{app}}] = 4\pi^2\alpha^2 \left(-\nu - \frac{d}{2}\right) \sum_{k\in\mathbb{Z}_N^d} \frac{\varphi(k)}{(1-\varphi(k))^2} \frac{g_{ij}^{(k)}}{a^{(k)}} v_k^T (C^{\text{app}})^{-1} u_k$$

which yields the result. □

For the Gaussian-like DPP prior, from equations (4.23) and (4.26), equation (4.22) results

$$\nabla \log f_{\text{DPP}}^{\text{app}}(\boldsymbol{\mu} \mid \Lambda) = (2\pi^2 c^{-\frac{2}{d}}) \sum_{k\in\mathbb{Z}_N^d} g^{(k)} \frac{\varphi(k)}{(1-\varphi(k))^2} \left[\frac{1-\varphi(k)}{1-e^{-D^{\text{app}}}} - v_k^T (C^{\text{app}})^{-1} u_k\right]$$

Similarly, for the Whittle-Matern-like DPP prior, from equations (4.24) and (4.27), equation (4.22) results

$$\nabla \log f_{\text{DPP}}^{\text{app}}(\boldsymbol{\mu} \mid \Lambda) = 4\pi^2\alpha^2 \left(\nu + \frac{d}{2}\right) \sum_{k\in\mathbb{Z}_N^d} \frac{g^{(k)}}{a^{(k)}} \frac{\varphi(k)}{(1-\varphi(k))^2} \left[\frac{1-\varphi(k)}{1-e^{-D^{\text{app}}}} - v_k^T (C^{\text{app}})^{-1} u_k\right]$$

which concludes the proof

□

4.B Measure-Theoretic Details

From the discussion in the main text, we have that $\boldsymbol{y} \in \mathbb{R}^{p\times n}$, $\boldsymbol{\eta} \in \mathbb{R}^{d\times n}$, $\Sigma \in \mathbb{R}_+^p$, $\Lambda \in \mathbb{R}^{p\times d}$, $\boldsymbol{\psi} \in \mathbb{R}_+^{p\times d}$, $\boldsymbol{\phi} \in \mathbb{S}^{p\times d-1}$ (the $pd-1$ dimensional simplex), and $\tau \in \mathbb{R}_+$. Moreover, we consider $\boldsymbol{\mu}$ as a random point configuration, which takes values in $\Omega = \cup_{m=0}^{\infty}\Omega_m$ where Ω_m denotes the space of (pairwise distinct) m-uples of $\mathbb{R}^d$. We endow each Ω_m with the smallest σ-algebra which makes the following mapping measurable

$$(\mu_1, \dots, \mu_m) \mapsto \{\mu_1, \dots, \mu_m\}.$$

where on the left hand side we see $\mu_1, \dots, \mu_m$ as an ordered vector in R^m and on the right hand side as an unordered collection of points in R, where $R \subset \mathbb{R}^d$ is the hypersquare where μ is defined. The σ-algebra on Ω is then the smallest σ-algebra containing the union of all the σ-algebras on each Ω_m. Then, it follows that $(\boldsymbol{\mu}, \boldsymbol{s}, \boldsymbol{\Delta}, \boldsymbol{c}) \in \cup_{m=0}^{\infty}\left\{\Omega_m \times \mathbb{R}_+^m \times \mathcal{SP}_d^m \times \{1, \dots, m\}^n\right\}$, where $\mathcal{SP}_d$ denotes the space of $d\times d$ symmetric and positive matrices.

Consider now sets $\mathfrak{Y} \subset \mathbb{R}^{p\times n}$, $\mathfrak{N} \subset R^{d\times n}$, $\Xi \subset \mathbb{R}^p_+$, $\Lambda \subset \mathbb{R}^{p\times d}$, $\Psi \subset \mathbb{R}^{p\times d}_+$, $\Phi \subset \mathbb{S}^{p\times d-1}$ $\mathfrak{T} \subset \mathbb{R}_+$, $\mathfrak{O}_m \subset \Omega_m$, $\mathfrak{S}_m \subset \mathbb{R}^m_+$, $\mathfrak{D}_m \subset \mathcal{SP}^m_d$, $\mathfrak{C}_m \subset \{1, \dots, m\}^n$. The dominating measure for the joint distribution of data and parameters is

$$\begin{aligned}&\nu\left(\mathfrak{Y} \times \mathfrak{N} \times \times \Xi \times \Lambda \times \Psi \times \Phi \times \mathfrak{T} \times \cup_{m\geq 0} \left\{\mathfrak{O}_m \times \mathfrak{S}_m \times \mathfrak{D}_m \times \mathfrak{C}_m\right\}\right) = \\ &\quad\int_{\mathfrak{Y}} \mathrm{d}\boldsymbol{y} \times \int_{\mathfrak{n}} \mathrm{d}\boldsymbol{\eta} \times \int_{\Xi} \mathrm{d}\boldsymbol{\Sigma} \times \int_{\Lambda} \mathrm{d}\Lambda \times \int_{\Psi} \mathrm{d}\boldsymbol{\psi} \times \int_{\Phi} \mathrm{d}\boldsymbol{\phi} \times \int_{\mathfrak{T}} \mathrm{d}\tau \times \\ &\qquad\qquad \times \sum_{m=0}^{\infty} \frac{e^{-|R|}}{m!} \int_{\mathfrak{O}_m} \mathrm{d}\mu_m \times \int_{\mathfrak{S}_m} \mathrm{d}s_m \times \int_{\mathfrak{D}_m} \mathrm{d}\Delta_m \times \sum_{c_1,\dots,c_n=1}^{M} \mathbb{1}[\boldsymbol{c} \in \mathfrak{C}_m]\end{aligned}$$

The density of data and parameters with respect to ν is:

$$\begin{aligned}&p(\boldsymbol{y}, \boldsymbol{c}, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{S}, \boldsymbol{\Delta}, \Sigma, \Lambda, \boldsymbol{\psi}, \boldsymbol{\phi}, \tau) = \\ &\quad \frac{1}{T^n}\left[\prod_{i=1}^{n} \mathcal{N}_p(y_i \,|\, \Lambda\eta_i, \Sigma)\right] \left[\prod_{h=1}^{k} (S_h^{(a)})^{n_h} \mathrm{Ga}(S_h^{(a)} \,|\, \alpha, 1) \mathrm{IW}(\Delta_h^{(a)} \,|\, \nu_0, \Psi_0) \prod_{i:c_i=h} \mathcal{N}_d(\eta_i \,|\, \mu_h^{(a)}, \Delta_h^{(a)})\right] \\ &\quad \left[\prod_{h=1}^{\ell} \mathrm{Ga}(S_h^{(na)} \,|\, \alpha, 1) \mathrm{IW}(\Delta_h^{(na)} \,|\, \nu_0, \Psi_0)\right] f^{\mathrm{app}}_{\mathrm{DPP}}(\boldsymbol{\mu}^{(a)} \cup \boldsymbol{\mu}^{(na)} \,|\, \rho, \Lambda, K_0; R) \\ &\quad \left[\prod_{j=1}^{p} \text{inv-Ga}(\sigma_j^2 \,|\, a_\sigma, b_\sigma) \prod_{h=1}^{d} \mathcal{N}(\lambda_{jh} \,|\, 0, \psi_{jh}\phi_{jh}^2\tau) \mathrm{Exp}(\Psi_{jw} \,|\, 1/2)\right] \mathrm{Dir}(vec(\phi) \,|\, a) \mathrm{Ga}(\tau \,|\, pda, 1/2)\end{aligned}$$

We now introduce the auxiliary variable u such that $u \,|\, T \sim \mathrm{Ga}(n, t)$ and consider the extended parameter space including $u \in \mathbb{R}_+$. Moreover, conditional to $\boldsymbol{c}$ we split $\boldsymbol{\mu} = \mu^{(a)} \cup \mu^{(na)}$, $\boldsymbol{S} = [\boldsymbol{S}^{(a)}, \boldsymbol{S}^{(na)}]$ and $\boldsymbol{\Delta} = [\boldsymbol{\Delta}^{(a)}, \boldsymbol{\Delta}^{(na)}]$ into allocated and non-allocated components (denoted with the (a) and (na) superscript respectively). The dominating measure ν' on the extended space can be straightforwardly derived. See, for instance, Equation (17) in Beraha et al. (2022).

4.C Additional simulations

In figure 4.C.1, we report the comparison, in terms of memory usage (measured in Bytes) and execution time per iteration (measured in seconds), between the AD approach and the analytical approach in sampling the high-dimensional matrix of loadings Λ. Fixing all the other model parameters, we set $N = 4$ and 6 component centers μ_h's (this choice impacts on both the memory usage and the execution time) and we compare the performance in sampling only the matrix Λ. We use 100 data points simulated from a p-dimensional Gaussian distribution, for $p = 100, 200$.

In figure 4.C.2, we report the chain of the number of clusters produced by the Lamb algorithm along the iterations. We perform 10^5 iterations and we plot one every ten iterations. We use the data of simulation study B, with $p = 100$, $d = 5$ and set $\alpha_{DP} = 0.5$. Note that the chain has not reached convergence yet after 10^5 iterations, demanding a very long burn-in phase. The poor efficiency of the Lamb algorithm is common in all the settings analyzed in our simulation studies.

Tables 4.C.1 - 4.C.6 report the summary statistics for the clustering in the two simulations discussed in Section 4.4.

			MODE NCLUS	MEAN NCLUS	ARI BEST	CI ARI
d	Model	Parameter				
2	Lamb	0.1	5	4.76	0.94	[0.953, 0.955]
		0.5	5	5.26	0.94	[0.932, 0.934]
		1.0	28	27.61	0.22	[0.268, 0.27]
	APPLAM	5.0	6	6.64	0.83	[0.824, 0.829]
		10.0	9	9.02	0.72	[0.717, 0.725]
		20.0	6	6.07	0.91	[0.883, 0.889]
5	Lamb	0.1	54	54.27	0.09	[0.099, 0.1]
		0.5	62	62.65	0.08	[0.081, 0.081]
		1.0	41	39.70	0.34	[0.354, 0.355]
	APPLAM	5.0	4	4.01	1.00	[1.0, 1.0]
		10.0	4	4.02	1.00	[1.0, 1.0]
		20.0	4	4.00	1.00	[1.0, 1.0]
8	Lamb	0.1	38	39.17	0.73	[0.734, 0.735]
		0.5	30	30.42	0.68	[0.67, 0.67]
		1.0	31	31.00	0.75	[0.751, 0.751]
	APPLAM	5.0	4	4.02	1.00	[1.0, 1.0]
		10.0	4	4.00	1.00	[1.0, 1.0]
		20.0	4	4.00	1.00	[1.0, 1.0]

Table 4.C.1: Simulation study A, $p = 100$: comparison on the posterior number of clusters and on the quality of the inferred clusterings.

			MODE NCLUS	MEAN NCLUS	ARI BEST	CI ARI
d	Model	Parameter				
2	Lamb	0.1	6	5.85	0.92	[0.901, 0.905]
		0.5	27	27.20	0.27	[0.254, 0.256]
		1.0	30	29.79	0.24	[0.238, 0.239]
	APPLAM	5.0	6	6.28	0.86	[0.822, 0.825]
		10.0	6	6.41	0.84	[0.837, 0.84]
		20.0	7	6.92	0.85	[0.837, 0.841]
5	Lamb	0.1	66	68.96	0.06	[0.068, 0.068]
		0.5	69	70.11	0.06	[0.068, 0.069]
		1.0	65	68.00	0.06	[0.072, 0.072]
	APPLAM	5.0	4	4.01	1.00	[1.0, 1.0]
		10.0	4	4.01	1.00	[1.0, 1.0]
		20.0	4	4.00	1.00	[1.0, 1.0]
8	Lamb	0.1	39	39.93	0.23	[0.228, 0.229]
		0.5	38	40.82	0.23	[0.238, 0.239]
		1.0	62	61.84	0.11	[0.108, 0.109]
	APPLAM	5.0	4	4.00	1.00	[1.0, 1.0]
		10.0	4	4.00	1.00	[1.0, 1.0]
		20.0	4	4.00	1.00	[1.0, 1.0]

Table 4.C.2: Simulation study A, $p = 200$: comparison on the posterior number of clusters and on the quality of the inferred clusterings.

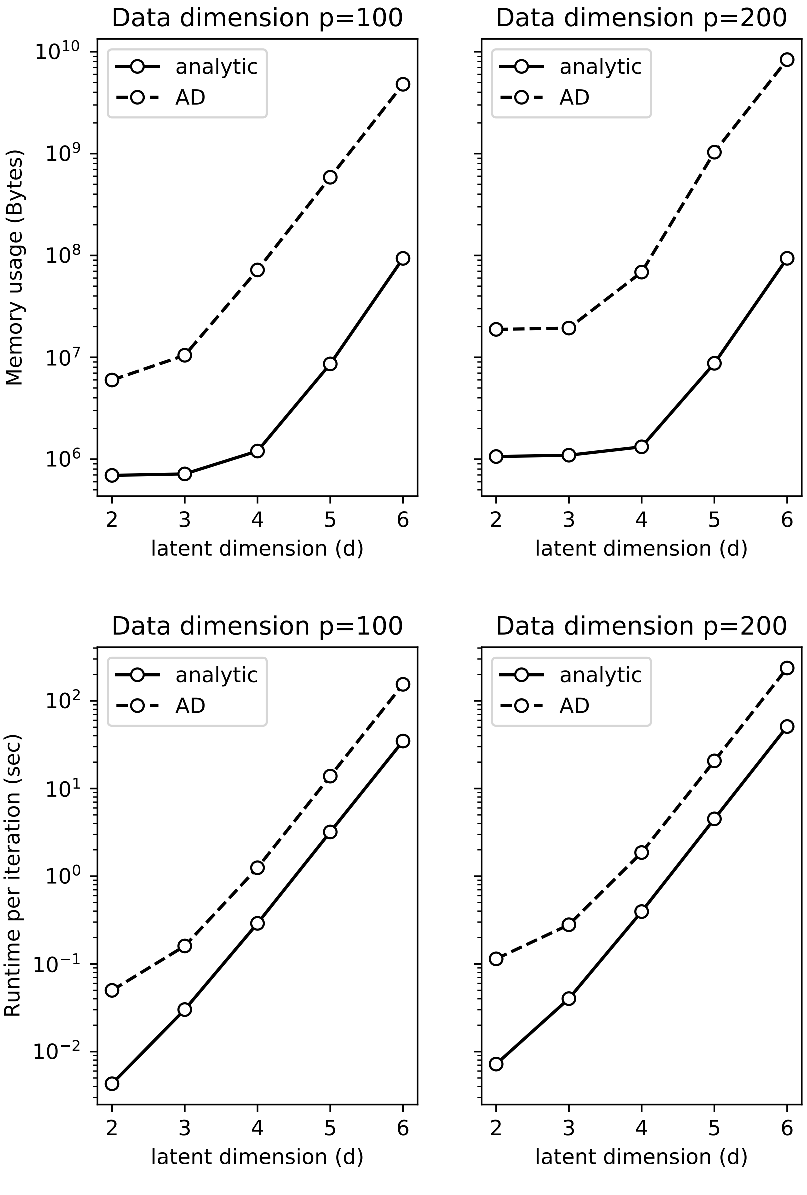


Figure 4.C.1: Memory requirement (top row) and run-time execution per iteration of MCMC with $n = 100$ samples when the data-dimension is $p = 100$ (left plot) and $p = 200$ (right plot) as the latent dimension d varies.

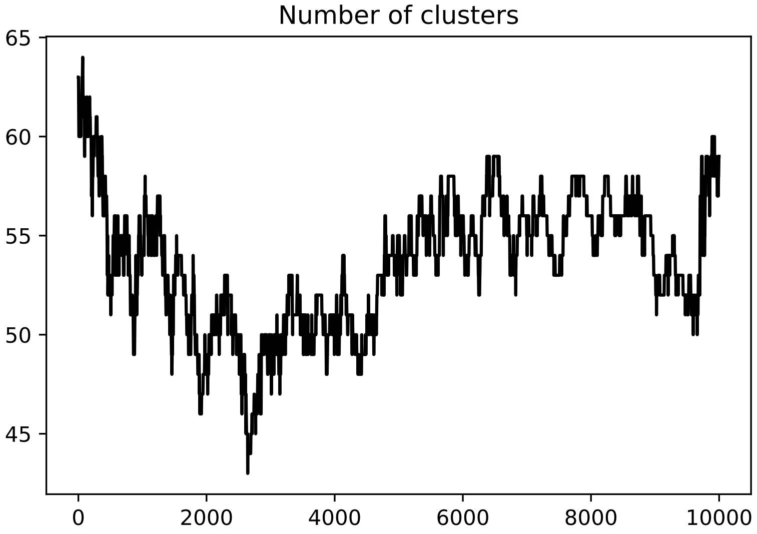


Figure 4.C.2: Chain of the number of clusters produced by the Lamb algorithm along 10^5 iterations. Data come from the simulation study B, with $p = 100$, $d = 5$ and we set $\alpha_{DP} = 0.5$

			MODE NCLUS	MEAN NCLUS	ARI BEST	CI ARI
d	Model	Parameter				
2	Lamb	0.1	10	9.97	0.72	[0.716, 0.72]
		0.5	37	37.02	0.19	[0.194, 0.195]
		1.0	39	38.98	0.20	[0.185, 0.186]
	APPLAM	5.0	6	5.67	0.60	[0.596, 0.599]
		10.0	8	7.86	0.79	[0.795, 0.799]
		20.0	6	5.99	0.80	[0.709, 0.721]
5	Lamb	0.1	84	83.43	0.05	[0.051, 0.051]
		0.5	85	85.25	0.05	[0.051, 0.051]
		1.0	86	86.09	0.05	[0.05, 0.051]
	APPLAM	5.0	4	3.77	1.00	[0.852, 0.873]
		10.0	4	4.36	1.00	[0.99, 0.992]
		20.0	4	4.12	1.00	[0.989, 0.991]
8	Lamb	0.1	75	74.37	0.07	[0.075, 0.075]
		0.5	73	72.34	0.08	[0.079, 0.079]
		1.0	76	76.51	0.08	[0.083, 0.083]
	APPLAM	5.0	4	4.00	1.00	[1.0, 1.0]
		10.0	4	4.00	1.00	[1.0, 1.0]
		20.0	4	4.00	1.00	[1.0, 1.0]

Table 4.C.3: Simulation study A, $p = 400$: comparison on the posterior number of clusters and on the quality of the inferred clusterings.

			mode_nclus	avg_nclus	ari_best_clus	CI_aris
Latent dim	Model	Parameter				
2	Lamb	0.1	7	7.29	0.87	[0.857, 0.858]
		0.5	8	7.90	0.87	[0.858, 0.859]
		1.0	8	8.29	0.88	[0.86, 0.861]
	APPLAM	5.0	8	8.53	0.68	[0.657, 0.662]
		10.0	11	11.12	0.50	[0.513, 0.518]
		20.0	9	9.50	0.66	[0.633, 0.639]
5	Lamb	0.1	65	63.71	0.08	[0.083, 0.083]
		0.5	68	68.18	0.08	[0.08, 0.08]
		1.0	69	68.98	0.07	[0.079, 0.079]
	APPLAM	5.0	6	6.02	0.95	[0.948, 0.949]
		10.0	4	4.35	0.95	[0.946, 0.946]
		20.0	5	4.88	0.97	[0.964, 0.965]
8	Lamb	0.1	30	29.59	0.76	[0.764, 0.764]
		0.5	40	40.00	0.68	[1.0, 1.0]
		1.0	34	34.45	0.74	[0.738, 0.738]
	APPLAM	5.0	3	3.48	0.67	[0.658, 0.661]
		10.0	4	4.38	1.00	[0.993, 0.994]
		20.0	5	5.00	0.97	[0.972, 0.973]

Table 4.C.4: Simulation study B, $p = 100$: comparison on the posterior number of clusters and on the quality of the inferred clusterings.

			mode_nclus	avg_nclus	ari_best_clus	CI_aris
Latent dim	Model	Parameter				
2	Lamb	0.1	9	8.45	0.88	[0.86, 0.861]
		0.5	9	9.03	0.88	[0.859, 0.86]
		1.0	9	9.44	0.88	[0.855, 0.856]
	APPLAM	5.0	8	7.90	0.69	[0.64, 0.647]
		10.0	7	6.87	0.61	[0.589, 0.594]
		20.0	7	7.23	0.69	[0.682, 0.684]
5	Lamb	0.1	8	8.22	0.97	[0.947, 0.949]
		0.5	75	75.26	0.06	[0.071, 0.071]
		1.0	76	76.42	0.07	[0.069, 0.069]
	APPLAM	5.0	4	4.47	0.95	[0.946, 0.947]
		10.0	6	5.90	0.91	[0.893, 0.897]
		20.0	5	4.73	0.93	[0.935, 0.936]
8	Lamb	0.1	61	60.44	0.16	[0.155, 0.155]
		0.5	62	61.59	0.13	[0.133, 0.133]
		1.0	61	60.54	0.16	[0.155, 0.155]
	APPLAM	5.0	4	4.07	0.96	[0.957, 0.958]
		10.0	4	4.00	0.97	[0.97, 0.971]
		20.0	4	4.00	0.99	[0.987, 0.987]

Table 4.C.5: Simulation study B, $p = 200$: comparison on the posterior number of clusters and on the quality of the inferred clusterings.

			mode_nclus	avg_nclus	ari_best_clus	CI_aris
Latent dim	Model	Parameter				
2	Lamb	0.1	9	9.72	0.87	[0.849, 0.851]
		0.5	11	10.81	0.85	[0.832, 0.834]
		1.0	11	11.29	0.84	[0.823, 0.825]
	APPLAM	5.0	6	6.52	0.46	[0.462, 0.464]
		10.0	12	13.04	0.63	[0.63, 0.633]
		20.0	7	6.91	0.46	[0.463, 0.465]
5	Lamb	0.1	8	8.17	0.97	[0.95, 0.951]
		0.5	8	8.32	0.97	[0.948, 0.95]
		1.0	91	91.02	0.05	[0.055, 0.055]
	APPLAM	5.0	6	6.35	0.92	[0.91, 0.911]
		10.0	6	5.82	0.91	[0.911, 0.913]
		20.0	5	5.15	0.91	[0.918, 0.919]
8	Lamb	0.1	77	78.12	0.09	[0.093, 0.094]
		0.5	75	75.02	0.11	[0.106, 0.106]
		1.0	84	83.47	0.08	[0.083, 0.083]
	APPLAM	5.0	4	4.01	0.97	[0.973, 0.973]
		10.0	5	4.73	0.95	[0.96, 0.961]
		20.0	4	4.07	0.97	[0.974, 0.975]

Table 4.C.6: Simulation study B, $p = 400$: comparison on the posterior number of clusters and on the quality of the inferred clusterings.

5. Dependent Random Probability Measures for Bayesian inference

The second part of this thesis is dedicated to the study of dependent random probability measures and their applications in Bayesian nonparametric statistics. In particular, we consider the setting in which observations y_i's are associated with covariates. When the covariates take only a finite number of values, the unique values in the covariates identify groups of observations, that is, subpopulations of data that share the same value of covariates. Within this scenario, Chapter 6, based on Beraha et al. (2021), joint work with Alessandra Guglielmi and Fernando A. Quintana is concerned with detecting homogeneity of distributions and identifying clusters of homogeneous populations. Chapter 7, based on Beraha et al. (2021), joint work with Matteo Pegoraro, Riccardo Peli, and Alessandra Guglielmi, deals with spatially referenced data, where each group is associated with a specific geographical area, with the aim of estimating the data-genarating density in each group taking into account for spatial dependence. In Chapter 8, based on Beraha and Griffin (2022), joint work with Jim. E. Griffin, we consider a different problem: in addition to modeling the data in each group, we also want to explore and represent the difference in distribution across subpopulations, obtaining low-dimensional and interpretable summaries. In Chapter 9 we consider a more general setting, where vector-valued covariates are recorded and, typically, each observation is associated with a different covariate value. Chapter 9, based on Beraha et al. (2022), joint work with Alessandra Guglielmi, Fernando A. Quintana, Maria de Iorio, Johan Gunnar Eriksson, and Fabian Yap presents an application to growth curves of kids in Singapore, whose height and weight is recorded from birth to the age of 7 years.

5.1 Departures from exchangeability

In Chapter 1, exchangeability has been a key motivation for the development of Bayesian nonparametric approaches for clustering and density modeling. In fact, for observations $y_1, \ldots, y_n$, de Finetti's representation theorem ensures the existence of a "likelihood function" $\mathsf{P}(y_i \in \mathrm{d}x_i \mid \nu)$, such that, given a parameter ν, data are conditionally i.i.d. from the likelihood, and of a "prior distribution" $Q(\mathrm{d}\nu)$ for the parameter ν, therefore fully motivating the Bayesian approach.

Given the strength of this result, it should not come as a surprise that exchangeability holds only in a few cases. For example, if y_i represents the average daily temperature of the i-th day, it is natural to exclude the possibility that the distribution of $y_1, \ldots, y_n$ might be invariant with respect to permutations. Similarly, the same applies if $y_1, \ldots, y_{n_1}$ measures a clinical quantity of a control group of patients and $y_{n_1+1}, \ldots, y_n$ measures the same quantity in a response group. More generally, whenever covariates are associated with observations, we must be wary of the exchangeability assumption.

This section introduces weaker definitions of exchangeability that we might expect to hold in different practical situations and with the associated de Finetti-type representation results.

5.1.1 Partial exchangeability

Start by assuming that a single covariate $g_i \in \{1, \dots, I\}$ is associated with each observation. That is, we can consider observations $\{y_{i,j}\}_{i,j}$ for $i = 1, \dots, I$ and $j = 1, \dots, n_i$ to be divided into subpopulations or groups.

To be mathematically accurate, let us introduce partial exchangeability for a sequence of random variables. Let $\mathbb{Y}$ denote a complete and separable metric space (i.e., a Polish space) with the corresponding metric d. Let $\mathcal{Y}$ denote the Borel σ-algebra of $\mathbb{Y}$, and $\mathbb{P}_{\mathbb{Y}}$ denote the space of all probability measures on $(\mathbb{Y}, \mathcal{Y})$, with Borel σ-algebra $\mathcal{P}_{\mathbb{Y}}$. We will often skip reference to σ-algebras. A double sequence $(y_{11}, y_{12}, y_{13}, \dots, y_{21}, y_{22}, y_{23}, \dots)$ of $\mathbb{Y}$-valued random variables, defined on a probability space $(\Omega, \mathcal{F}, \mathsf{P})$ is called *partially exchangeable* if for all $n, m \geq 1$ and all permutations $(i(1), \dots, i(n))$ and $(j(1), \dots, j(m))$ of $(1, \dots, n)$ and $(1, \dots, m)$, respectively, we have

$$\mathcal{L}(y_{11}, \dots, y_{1n}, y_{21}, \dots, y_{2m}) = \mathcal{L}(y_{1i(1)}, \dots, y_{1i(n)}, y_{2j(1)}, \dots, y_{2j(m)}).$$

Thus, partial exchangeability can be conceptualized as invariance of the joint law above under the class of *all* permutations acting on the indices *within* each of the samples. Here and from now on, the distribution of a random element y is denoted by $\mathcal{L}(y)$.

The previous setting can be immediately extended to the case of I different populations or groups. By de Finetti's representation theorem (see the proof in Regazzini, 1991), partial exchangeability for the array of I sequences of random variables $(y_{11}, y_{12}, \dots, y_{21}, y_{22}, \dots, y_{I1}, y_{I2}, \dots)$ is equivalent to

$$\mathsf{P}(y_{ij} \in A_{ij},\, j = 1, \dots, N_i,\, i = 1, \dots, I) = \int_{\mathbb{P}_{\mathbb{Y}}^I} \prod_{i=1}^{I} \prod_{j=1}^{N_i} p_i(A_{ij})\, Q(dp_1, \dots, dp_I),$$

for any $N_1, \dots, N_I \geq 1$ and Borel sets $\{A_{ij}\}$ for $j = 1, \dots, N_i$ and $i = 1, \dots, I$. In this case, the de Finetti measure Q is defined on the I-fold product space $\mathbb{P}_{\mathbb{Y}}^I = \mathbb{P}_{\mathbb{Y}} \times \mathbb{P}_{\mathbb{Y}} \times \cdots \times \mathbb{P}_{\mathbb{Y}}$, and $(p_1, p_2, \dots, p_I) \sim Q$. The entire joint sequence of random variables is exchangeable if and only if Q gives probability 1 to the measurable set $S = \{(p_1, p_2, \dots, p_I) \in \mathbb{P}_{\mathbb{Y}}^I : p_1 = p_2 = \cdots = p_I\}$.

5.1.2 Separate exchangeability

Consider now a 2-array y_{ij}, $(i, j) \in \mathbb{N} \times \mathbb{N}$. We say that $\{y_{ij}\}_{ij}$ is separately (or jointly) exchangeable if for any finite permutations σ, π of $\mathbb{N}$, we have

$$\mathcal{L}((y_{ij})_{ij}) = \mathcal{L}((y_{\sigma(i)\,\pi(j)})_{ij}),$$

that is, the distribution of the y_{ij}'s is invariant under separate permutations σ and π of rows and columns respectively.

In Bayesian nonparametrics, separate exchangeability has been discussed in the context of network data, where y_{ij} represents a binary adjacency matrix, in Caron and Fox (2017), Orbanz and Roy (2014) and several later works. More recently, Lin et al. (2021) proposed to replace the partial exchangeability assumption with the separate exchangeability one also when considering grouped data, to account for the possibility of observations referring to the same individual appearing in different groups. That is the case, for instance, where observations are patients and groups are hospitals, and some patients have been treated in more than one hospital.

The Aldous-Hoover theorem (see Theorem 28.2 in Kallenberg, 2021) entails the existence of a random measurable function $f : [0, 1]^4 \to \mathbb{R}$ and uniformly distributed variables α, ξ_i and $\zeta_{ij} = \zeta_{ji}$ such that

$$y_{ij} = f(\alpha, \xi_i, \xi_j, \zeta_{ij}).$$

5.1.3 Local exchangeability

The definitions of partial and separate exchangeability allow only for data clearly divided into groups, i.e., when a single categorical covariate is considered. In real applications, it is often the case that data are associated with (vector-valued) covariates $\boldsymbol{x}_i$ taking values in a possibly uncountable set $\mathbb{X}$, and that $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ for $i \neq j$.

Local exchangeability (Campbell et al., 2019) interpolates between classical exchangeability and partial exchangeability by prescribing that permuting observations y_i associated with similar values of covariates produces a small deviation in the joint law of the data. Formally, let d be a premetric over $\mathbb{X}$ (i.e., a function $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_+$ such that $d(x, y) \geq 0$ and $d(x, x) = 0$ for all $x, y \in \mathbb{X}$), and denode with d_{TV} the total variation metric. Then $(y_i)_{i\geq 1}$ with associated covariates $(\boldsymbol{x}_i)_{i\geq 1}$ is locally exchangeable if

$$d_{TV}\left(\mathcal{L}(y_1, y_2, \ldots), \mathcal{L}(y_{\sigma(1)}, y_{\sigma(1)}, \ldots)\right) \leq \sum_{i\geq 1} d(\boldsymbol{x}_i, \boldsymbol{x}_{\sigma(i)})$$

for any finite permutation σ. Setting $d \equiv 0$ recovers the usual notion of exchangeability, while $d(\boldsymbol{x}, \boldsymbol{x}') = I[\boldsymbol{x} \neq \boldsymbol{x}']$ yields partial exchangeability where the groups are identified by the unique values in $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$.

Under some conditions on $(\mathbb{X}, d)$, Campbell et al. (2019) show that a de Finetti representation holds; see their Theorem 5.

5.2 Priors for dependent random probability measures

Vectors of dependent random distributions appeared first in Cifarelli and Regazzini (1978), but it was in MacEachern (1999) where a large class of dependent Dirichlet processes was introduced. We give a succinct overview of some constructions next. To this end, it is useful to recall here the basic Dirichlet process model, see Chapter 1 for further details.

$$\begin{aligned} y_i \mid \tilde{p} &\overset{\text{iid}}{\sim} \tilde{p}, \qquad i = 1, \ldots, n \\ \tilde{p} &= \sum_{h\geq 1} w_h \delta_{\theta_h} \sim DP(\alpha, G_0) \end{aligned} \tag{5.1}$$

where G_0 is a diffuse probability measure on the space $\mathbb{Y}$ endowed with its Borel σ-field. Dependent random distributions have been active areas of research in the last 20 years, so that a comprehensive review is beyond the scope of this thesis, and we just limit ourselves to giving the basic definitions and providing background material for the following chapters. We refer to Quintana et al. (2022) for a detailed review.

5.2.1 The dependent Dirichlet process

We start by considering the most general setting, where each observation y_i is associated with a (vector-valued) covariate $\boldsymbol{x}_i \in \mathbb{X}$. The natural extension to (5.1) is to assume

$$y_i \mid \tilde{p}_{\boldsymbol{x}_i} \overset{\text{ind}}{\sim} \tilde{p}_{\boldsymbol{x}_i}, \qquad i = 1, \ldots, n. \tag{5.2}$$

Therefore, instead of a single probability measure $\tilde{p}$, the task now is to model a collection of probability measures $\{\tilde{p}_{\boldsymbol{x}}\}$ indexed by $\boldsymbol{x} \in \mathbb{X}$. A flexible class of models, termed the dependent Dirichlet process (DDP), was introduced in MacEachern (1999).

The key idea is to define a stochastic process over $\mathbb{P}_{\mathbb{Y}}$ (the space of probability measure over $\mathbb{Y}$) indexed by $\boldsymbol{x} \in \mathbb{X}$ such that marginally the random measure $\tilde{p}_{\boldsymbol{x}}$ is distributed as a Dirichlet process. This can be achieved by setting

$$\tilde{p}_{\boldsymbol{x}}(\cdot) = \sum_{h\geq 1} w_h(\boldsymbol{x}) \delta_{\theta_h(\boldsymbol{x})}(\cdot)$$

where $w_j(\boldsymbol{x}) = \nu_j(\boldsymbol{x}) \prod_{\ell<j}(1-\nu_\ell(\boldsymbol{x}))$ and for each $j \geq 1$, $\{\nu_j(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathbb{X}}$ is stochastic processes with Beta$(1,\alpha)$ marginals, independent of $\{\theta_j(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathbb{X}}$, a stochastic process with G_0 marginals. Moreover, and the processes $\{\nu_j(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathbb{X}}$, $\{\theta_j(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathbb{X}}$ are independent across different values of $j \geq 1$. Optionally α and G_0 could depend on $\boldsymbol{x}$ as well.

Theoretical properties of the DDP have been investigated in Barrientos et al. (2012). Campbell et al. (2019) showed that under suitable assumptions on the stochastic processes $\nu_j(\boldsymbol{x})$ and $\theta_j(\boldsymbol{x})$, local exchangeability holds under the DDP.

The general construction of MacEachern's DDP has been specialized and extended in several papers. One of the first ones is De Iorio et al. (2004), where the weights are assumed independent of $\boldsymbol{x}$ and the dependence on the covariates is assumed only through the atoms, in an ANOVA fashion for grouped data. The "single weight" construction was later replaced by several "single atoms" models. As noted in Quintana et al. (2022), assuming covariate-dependent weights leads to better predictive performances especially for non-observed values of the covariates. Moreover, it is usually easy to extend the model to encompass covariate-dependent atoms. See Quintana et al. (2022) for several examples of DDP models.

5.3 Priors for partially exchangeable data

Let us consider more in detail the case of partially exchangeable data. Several papers have investigated the choice of prior Q for a vector of dependent random probability measures.

5.3.1 Hierarchical Processes

A traditional (and fruitful) approach for modeling data arising from a collection of groups or related studies involves the construction of hierarchical random prior probability measures. One of the first such examples in the BNP literature, is the well-known hierarchical DP introduced in Teh et al. (2006). The HDP assumes that

$$\begin{aligned} y_{i,1},\dots,y_{i,n_i} \mid \tilde{p}_i &\overset{\text{iid}}{\sim} \tilde{p}_i, \qquad i=1,\dots I \\ \tilde{p}_1,\dots,\tilde{p}_I \mid \tilde{p}_0 &\overset{\text{iid}}{\sim} DP(\alpha\tilde{p}_0) \\ \tilde{p}_0 &\sim DP(\gamma G_0) \end{aligned} \tag{5.3}$$

where $\alpha, \gamma > 0$ and G_0 is a (usually diffuse) probability measure. The stick-breaking representation of the DP entails

$$\tilde{p}_0 = \sum_{h\geq 1} \pi_h \delta_{\tau_h}, \qquad \tilde{p}_i = \sum_{h\geq 1} w_{ih}\delta_{\theta_{ih}} \tag{5.4}$$

where the weights $(\pi_h)_{h\geq 1}$ and $(w_{ih})_{h\geq 1}$ come from a stick-breaking process with parameters γ and α, respectively, the atoms $(\tau_h)_{h\geq 1}$ are i.i.d. from G_0 and $\theta_{ih} \mid \tilde{p}_0 \overset{\text{iid}}{\sim} \tilde{p}_0$ for all i, h. Therefore, the set of atoms $\{\theta_{ih}\}_{h\geq 1}$ coincides with $\{\tau_h\}_{h\geq 1}$ for all i's, meaning that all random measures $\tilde{p}_i$ have the same support points. The marginal law of observations under the HDP can be interpreted by means of a generalizations of the Chinese restaurant process to multiple restaurants, where each group of data is associated with a restaurant, termed "Chinese restaurant franchise" in Teh et al. (2006).

5.3.2 Dependent Normalized Random Measures

Vectors of completely random measures (also called completely random vectors, CRVs for short) can be constructed to yield dependence between the different measures. Their normalization has been used as nonparametric prior in different works. We say that a vector of random measures $(\widetilde{\mu}_1, \dots, \widetilde{\mu}_I)$ on the Polish space (Θ, d) with Borel σ-algebra $\mathcal{B}(\Theta)$ is completely random if for any n and pairwise disjoint $A_1, \dots, A_n$, the vectors $\{(\widetilde{\mu}_1(A_j), \dots, \widetilde{\mu}_I(A_j))\}_{j=1}^n \in \mathbb{R}_+{}^I$ are independent. Usually (see, e.g., Leisen et al., 2013), it is assumed that

$$(\widetilde{\mu}_1(A), \dots, \widetilde{\mu}_I(A)) = \sum_{k \geq 1} (s_{1k}, \dots, s_{IK}) I[x_k \in A], \qquad A \in \mathcal{B}(\Theta),$$

so that, marginally, $\widetilde{\mu}_j(\cdot) = \sum_{k \geq 1} s_{jk} \delta_{x_k}(\cdot)$ is a CRM. The support points are shared across all the random measures. Moreover, a Poisson process representation holds: $\{(s_{1k}, \dots, s_{Ik}, x_k)\}_{k \geq 1}$ are the points of a Poisson point process over $\mathbb{R}_+^I \times \Theta$ with intensity measure $\nu(\mathrm{d}\boldsymbol{s}\mathrm{d}x) = \rho(\boldsymbol{s})\mathrm{d}s\alpha(\mathrm{d}x)$ where $\boldsymbol{s} = (s_1, \dots, s_I)$ and $\mathrm{d}\boldsymbol{s} = \mathrm{d}s_1 \cdots \mathrm{d}s_I$ denotes the I-fold product measure. A CRV is uniquely determined by its Laplace transform, for measurable $f_i : \Theta \to R_+$, $i = 1, \dots, I$

$$\mathbb{E}\left[\exp\left\{-\sum_{i=1}^I \int_\Theta f_i(z)\widetilde{\mu}_i(\mathrm{d}z)\right\}\right] = \exp\left\{-\int_{\mathbb{R}_+^I \times \Theta} 1 - e^{-\sum_{i=1}^I s_i f_i(x)} \nu(\mathrm{d}\boldsymbol{s}\mathrm{d}x)\right\}.$$

The marginal intensity of $\widetilde{\mu}_i$ is $\nu_i(\mathrm{d}s_i\mathrm{d}x)$ is defined as

$$\nu_i(B \times A) = \nu(\mathbb{R}_+^{i-1} \times B \times \mathbb{R}_+^{I-i} \times A), \qquad B \subset \mathbb{R}_+,\ A \in \mathcal{B}(\Theta).$$

To ensure that the CRV can be normalized, the Lévy intensity must satisfy

$$\int_{\mathbb{R}_+^I \times B} \|s\|^2 \nu(\mathrm{d}\boldsymbol{s}\mathrm{d}x) < +\infty, \qquad B \in \mathcal{B}(\Theta).$$

Then $(\tilde{p}_1, \dots, \tilde{p}_I)$, $\tilde{p}_i := \widetilde{\mu}_i / \widetilde{\mu}(\Theta)$, is a vector of random probability measures.

Several construction for a vector of random measures have been proposed. Working directly on the definition of intensity ν, Leisen and Lijoi (2011) proposed the use of Lévy copulas to induce dependence between random measures with fixed margins, while Leisen et al. (2013) propose a multivariate Lévy intensity yielding Gamma process marginals. Other approaches have been focused on modelling the $\widetilde{\mu}_i$'s directly, for instance by using additive processes, such as Lijoi et al. (2014b) and Griffin et al. (2013), where each $\widetilde{\mu}_i$ is obtained by superimposing two or more completely random measures, namely by setting

$$\widetilde{\mu}_i = \sum_{h=1}^H \gamma_{ih} \mu_h^*$$

where γ_{ih} is a binary indicator which may be random. Yet another possibility is based on hierarchical constructions generalizing the hierarchical DP; which has been investigated in Camerlenghi et al. (2019), Argiento et al. (2019) and Bassetti et al. (2020).

Compound random measures (CoRMs, Griffin and Leisen, 2017) have been recently proposed as a flexible and simple construction for dependent random measures. To define a CoRM, consider a CRM $\nu = \sum_{k \geq 1} J_k \delta_{\theta_k}$ and set

$$\widetilde{\mu}_i = \sum_{k \geq 1} m_{ik} J_k \delta_{\theta_k}$$

where $\boldsymbol{m}_k = (m_{1k}, \dots, m_{Ik}) \in \mathbb{R}_+^I$ are i.i.d. vectors for $k = 1, 2, \dots$. Depending on the Lévy intensity of ν and the distribution of the m_{hk}'s, a large number of well-known marginal processes can be recovered.

5.3.3 Other approaches

Several approaches based on Pólya trees have been proposed to model dependent random probability measures, especially in the case of two groups of data. For instance, Ma and Wong (2011) and Soriano and Ma (2017) propose the coupling optional Pólya tree prior, which jointly generates two dependent random distributions through a random-partition-and-assignment procedure similar to Pólya trees. The former paper consider both testing hypotheses from a global point of view, while the latter takes a local perspective on the two-sample hypothesis, detecting high resolution local differences. Also Chen and Hanson (2014) and Holmes et al. (2015) consider the two-sample testing problem, using a Pólya tree prior for the common distribution in the null, while the model for the alternative hypothesis assumes that the two population distributions are independent draws from the same Pólya tree prior. Their approaches differ in the way they specify the Pólya tree prior.

6. The semi-hierarchical Dirichlet process and its application to clustering homogeneous distributions

Assessing homogeneity of distributions is an old problem that has received considerable attention, especially in the nonparametric Bayesian literature. To this effect, in this chapter, based on Beraha et al. (2021), we propose the semi-hierarchical Dirichlet process, a novel hierarchical prior that extends the hierarchical Dirichlet process of Teh et al. (2006) and that avoids the degeneracy issues of nested processes recently described by Camerlenghi et al. (2019). We go beyond the simple yes/no answer to the homogeneity question and embed the proposed prior in a random partition model; this procedure allows us to give a more comprehensive response to the above question and in fact find groups of populations that are internally homogeneous when $I \geq 2$ such populations are considered. We study theoretical properties of the semi-hierarchical Dirichlet process and of the Bayes factor for the homogeneity test when $I = 2$. Extensive simulation studies and applications to educational data are also discussed.

6.1 Introduction

Our first contribution is the introduction of a novel class of nonparametric priors that, just as discussed in Camerlenghi et al. (2019), avoids the degeneracy issue of the nested Dirichlet process (NDP) of Rodriguez et al. (2008) that arises from the presence of shared atoms across populations. Indeed, Camerlenghi et al. (2019) showed that under the NDP, if two populations share at least one common latent variable in the mixture model, then the model identifies the corresponding distributions as completely equal. To overcome the degeneracy issue, they resort to a latent nested construction in terms of normalized random measures that adds a shared random measure to draws from the NDP. Instead, we use a variation of the hierarchical DP (HDP Teh et al., 2006), that we term the semi-HDP, but where the baseline distribution is itself a mixture of a DP and a non-atomic measure. We will show that this procedure solves the degeneracy problem as well. While relying on a different model, Lijoi et al. (2020) also propose to build on the HDP, combining it with the NDP, to overcome the degeneracy issue of nested processes.

Our second contribution is that the proposed model overcomes some of the practical and applied limitations of the latent nested approach by Camerlenghi et al. (2019). As pointed out in Beraha and Guglielmi (2019), the latent nested approach becomes computationally burdensome in the case of $I > 2$ populations. In contrast, implementing posterior inference for the semi-HDP prior does not require restrictions on I. We discuss in detail how to carry out posterior inference in the context of hierarchical models based on the semi-HDP.

A third contribution of this article is that we combine the proposed semi-HDP prior with a random partition model that allows different populations to be grouped in clusters that are internally homogeneous, i.e. arising from the same distribution. See an early discussion of this idea in the context of contingency tables in Quintana (1998). The far more general extension we aim for here is also useful from the applied viewpoint of finding out which, if any, of the I populations are internally homogeneous when homogeneity of the whole set does not hold. For the purpose of assessing global exchangeability, one may resort to discrepancy measures (Gelman et al., 1996); see also Catalano et al. (2021). In our approach, homogeneity corresponds to a point-null hypothesis about a discrete vector parameter, as we adopt a 'larger' model for the alternative hypothesis within which homogeneity is nested. We discuss the specific case of adopting Bayes factors for the proposed test within the partial exchangeability framework. We show that the Bayes factor for this test is immediately available, and derive some of its theoretical properties.

The rest of this chapter is organized as follows. Section 6.2 gives some additional background that is relevant for later developments, presents the semi-HDP prior (Section 6.2.2) and, in particular, it describes a food court of Chinese restaurants with private and shared areas metaphor (Section 6.2.3). Section 6.3 studies several theoretical properties of the semi-HDP such as support, moments, the corresponding partially exchangeable partition probability function (in a particular case) and specially how the degeneracy issue is overcome under this setting. Section 6.3.3 specializes the discussion to the related issue of testing homogeneity when $I = 2$ populations are present, and we study properties of the Bayes Factor for this test. Section 6.4 describes a computational strategy to implement posterior inference for the class of hierarchical models based on our proposed semi-HDP prior. Extensive simulations, with $I = 2$, 4 and 100 populations are presented in Section 6.5. An application to an educational data set is discussed in Section 6.6. The article concludes with a discussion in Section 6.7. A The appendix collects the proofs for the theoretical results, together with additional formulas and figures, and a discussion on consistency for the Bayes Factor in the case of $I = 2$ homogeneous populations. Code for posterior inference has been implemented in `C++` and is available as part of the BayesMix library at https://github.com/bayesmix-dev/bayesmix.

6.2 Assessing Exchangeability within a Partially Exchangeable Framework

While exchangeability can be explored in more generality, for clarity of exposition we set up our discussion in the context of continuous univariate responses, but extensions to, e.g. multivariate responses, can be straightforwardly accommodated in our framework.

6.2.1 A common home for exchangeability and partial exchangeability

A flexible nonparametric model for each group can be constructed by assuming a mixture, where the mixing group-specific distribution $\tilde{p}_i$ is a random discrete probability measure (r.p.m.), i.e.

$$y_{ij} \,|\, \tilde{p}_i \overset{\text{iid}}{\sim} p_i(\cdot) = \int_\Theta k(\cdot \,|\, \theta)\, \tilde{p}_i(d\theta), \qquad j = 1, \ldots, N_i, \tag{6.1}$$

where $k(\cdot \,|\, \theta)$ is a density in $\mathbb{Y}$ for any $\theta \in \Theta$, and $\tilde{p}_i$ is, for example, a DP on Θ. Note that, with a little abuse of notation, p_i in (6.1) and in the rest of the chapter denotes the conditional population density of group i (before p_i represented the population distribution of group i in de Finetti's theorem). In what follows, we will always assume that the parametric space is contained in $\mathbb{R}^p$ for some positive integer p, and we will always assume the Borel σ–field $\mathcal{B}(\Theta)$ of Θ. Using the well-known alternative representation of the mixture in terms of latent variables, the previous expression is equivalent to assuming that for any i,

$$y_{ij} \,|\, \theta_{ij} \overset{\text{ind}}{\sim} k(\cdot \,|\, \theta_{ij}), \quad \theta_{ij} \,|\, \tilde{p}_i \overset{\text{iid}}{\sim} \tilde{p}_i, \quad j = 1, \ldots, N_i. \tag{6.2}$$

In this case, partial exchangeability of observations $(y_{ij})_{ij}$ is equivalent to partial exchangeability of the latent variables $(\theta_{ij})_{ij}$. Hence exchangeability of observations $(y_{ij})_{ij}$ is equivalent to the statement $\tilde{p}_1 = \tilde{p}_2 = \cdots = \tilde{p}_I$ with probability one.

In the next subsection we develop one of the main contributions of this chapter, namely, the construction of a prior distribution $\pi(\tilde{p}_1, \ldots, \tilde{p}_I)$ such that there is positive prior probability that $\tilde{p}_1 = \tilde{p}_2 = \cdots = \tilde{p}_I$, but avoiding the degeneracy issues discussed in Camerlenghi et al. (2019) and that would arise if we assumed that $(\tilde{p}_1, \ldots, \tilde{p}_I)$ were distributed as the NDP by Rodriguez et al. (2008). Briefly, $(\tilde{p}_1, \ldots, \tilde{p}_I)$ is distributed as the NDP if

$$\tilde{p}_i \,|\, \tilde{p}_0 \overset{\text{iid}}{\sim} \tilde{p}_0 = \sum_{\ell=1}^{\infty} \pi_\ell \delta_{\tilde{p}^*_\ell}, \quad i = 1, \ldots, I \quad \text{and} \quad \tilde{p}^*_\ell \overset{\text{iid}}{\sim} DP(\gamma, G_0),$$

i.e., the independent atoms in $\tilde{p}_0$ are all drawn from a DP on Θ, specifically $\tilde{p}^*_\ell = \sum_{h=1}^{\infty} w_{h\ell} \delta_{\theta_{h\ell}}$, with $\theta_{h\ell} \overset{\text{iid}}{\sim} G_0$, a probability measure on Θ, and $\alpha, \gamma > 0$. The weights $(\pi_j)_j$ and $(w_{h\ell})_h$, $\ell = 1, 2, \ldots$, are independently obtained from the usual stick-breaking construction, with parameters α and γ, respectively. Here $\mathcal{D}_{\gamma G_0}$ denotes the Dirichlet measure, i.e. the distribution of a r.p.m. that is a DP with measure parameter γG_0. However, nesting discrete random probability measures produces degeneracy to the exchangeable case. As mentioned in Section 6.1, Camerlenghi et al. (2019) showed that the posterior distribution degenerates to the exchangeable case whenever a shared component is detected, i.e., the NDP does not allow for sharing clusters among non-homogeneous populations. The problem is shown to affect any construction that uses nesting, and not just the NDP.

To overcome the degeneracy issue, while retaining flexibility, Camerlenghi et al. (2019) proposed the so-called Latent Nested Nonparametric priors. These models involve a shared random measure that is added to the draws from a Nested Random Measure, hence accommodating for shared atoms. See also the discussion by Beraha and Guglielmi (2019). There are two key ideas in their model: (i) nesting discrete random probability measures as in the case of the NDP, and (ii) contaminating the population distributions with a common component as in Müller et al. (2004) and also, Lijoi et al. (2014a). The contamination aspect of the model yields dependence among population-specific random probability measures, and avoids the degeneracy issue pointed out by the authors, while the former accounts for testing homogeneity in multiple-sample problems. Their approach, however, becomes computationally burdensome in the case of $I > 2$ populations, and it is not clear how to extend their construction to allow for the desired additional analysis, i.e. assessing which, if any, of the I populations are internally homogeneous when homogeneity of the whole set does not hold.

6.2.2 The Model

We present now a hierarchical model that allows us to assess homogeneity, while avoiding the undesired degeneracy issues and which further enables us to construct a grouping of populations that are internally homogeneous. To do so we create a hierarchical representation of distributions that emulates the behavior arising from an exchangeable partition probability function (EPPF; Pitman, 2006) such as the Pólya urn. But the main difference with previous proposals to overcome degeneracy is that we now allow for different populations to arise from the same distribution, while simultaneously incorporating an additional mechanism for populations to explicitly differ from each other.

Denote $[I] = \{1, \ldots, I\}$. A partition $S_1, \ldots, S_k$ of $[I]$ can be described by cluster assignment indicators $\boldsymbol{c} = (c_1, \ldots, c_I)$ with $c_i = \ell$ iff $i \in S_\ell$. Assume this partition arises from a given EPPF. We introduce the following model for the latent variables in a mixture model such as (6.2). Let $\boldsymbol{y}_i := (y_{i1}, \ldots, y_{iN_i})$, for $i = 1, \ldots, I$. We assume that $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_I$, given all the population distributions $\tilde{p}_1, \ldots, \tilde{p}_I$ are independent, and furthermore arising from

$$y_{ij} \mid \tilde{q}_1, \dots, \tilde{q}_I, \boldsymbol{c} \overset{\text{iid}}{\sim} \int_{\Theta} k(\cdot \mid \theta)\, \tilde{q}_{c_i}(d\theta),\ j = 1, \dots, N_i, \quad \text{for all } i \tag{6.3}$$

$$\boldsymbol{c} \sim \pi_c(c_1, \dots, c_I) \tag{6.4}$$

$$\tilde{q}_1, \dots \tilde{q}_I \mid \tilde{p} \overset{\text{iid}}{\sim} DP(\alpha, \tilde{p}) \tag{6.5}$$

$$\tilde{p} \ = \kappa G_0 + (1-\kappa)\tilde{q}_0 \tag{6.6}$$

$$\tilde{q}_0 \sim DP(\gamma, G_{00}) \tag{6.7}$$

$$\kappa \sim Beta(a_\kappa, b_\kappa), \tag{6.8}$$

where $\alpha, \gamma > 0$. Thus the role of the population mixing distribution $\tilde{p}_i$ in (6.1) – or, equivalently, in (6.2) – is now played by $\tilde{q}_{c_i}$. Observe that $\tilde{q}_1, \dots, \tilde{q}_I$ in (6.5) play a role similar to the cluster specific parameters in more standard mixture models. Consider for example a case where $I = 4$ and $\boldsymbol{c} = (1, 2, 3, 1)$. Under the above setting, $\tilde{q}_1, \tilde{q}_2, \tilde{q}_3$ define a model for three different distributions, so that populations 1 and 4 share a common mixing distribution, and $\tilde{q}_4$ is never employed.

Equation (6.5) means that conditionally on $\tilde{p}$ each $\tilde{q}_k$ is an independent draw from a DP prior with mean parameter $\tilde{p}$ (and total mass α), i.e. $\tilde{q}_k$ is a discrete r.p.m. on $\Theta \subset \mathbb{R}^p$ for some positive integer p, with $\tilde{q}_k = \sum_{h \geq 1} w_{kh} \delta_{\theta^*_{kh}}$ where for any k the weights are independently generated from a stick-breaking process, $\{w_{kh}\}_h \overset{\text{iid}}{\sim} SB(\alpha)$, i.e.

$$w_{k1} = \beta_{k1}, \qquad w_{kh} = \beta_{ih} \prod_{j=1}^{h-1} (1 - \beta_{kj}) \ \text{ for } h = 2, 3, \dots, \qquad \beta_{ij} \overset{\text{iid}}{\sim} Beta(1, \alpha),$$

and $\{\theta^*_{kh}\}_h$, $\{\beta_{kh}\}_h$ are independent, with $\theta^*_{kh} \mid \tilde{p} \overset{\text{iid}}{\sim} \tilde{p}$. We assume the centering measure $\tilde{p}$ in (6.6) to be a *contaminated draw* $\tilde{q}_0$ from a DP prior, with centering measure G_{00}, with a fixed probability measure G_0. Both G_0 and G_{00} are assumed to be absolutely continuous (and hence non-atomic) probability measures defined on $(\Theta, \mathcal{B}(\Theta))$.

By (6.7), $\tilde{q}_0 = \sum_{h \geq 1} p_h \delta_{\tau_h}$, where $\{p_h\}_h \sim SB(\gamma)$, $\tau_h \overset{\text{iid}}{\sim} G_{00}$ are independent weights and location points. The model definition is completed by specifying $\pi_c(c_1, \dots, c_I)$. We assume that the c_i's are (conditionally) i.i.d. draws from a categorical distribution on $[I]$ with weights $\boldsymbol{\omega} = (\omega_1, \dots, \omega_I)$, i.e. $c_i \mid \boldsymbol{\omega} \overset{\text{iid}}{\sim} Cat([I]; \boldsymbol{\omega})$, where the elements of $\boldsymbol{\omega}$ are non-negative and constrained to add up to 1. A convenient prior for $\boldsymbol{\omega}$ is a finite dimensional Dirichlet distribution with parameter $\boldsymbol{\eta} = (\eta_1, \dots \eta_I)$. Observe that distributions $\tilde{q}_{c_1}, \dots, \tilde{q}_{c_I}$ allow us to cluster populations, so that there are at most I clusters and consequently $\tilde{q}_1, \dots, \tilde{q}_I$ are all of the cluster distributions that ever need to be considered.

We say that a vector of random probability measures $(\tilde{q}_1, \dots, \tilde{q}_I)$ has the semi-hierarchical Dirichlet process (semi-HDP) distribution if (6.5)-(6.7) hold, and we write $(\tilde{q}_1, \dots, \tilde{q}_I) \sim$ $semiHDP(\alpha, \gamma, \kappa, G_0, G_{00})$. It is straightforward to prove that, conditional on κ and eventual hyperparameters in G_0 and G_{00}, the expectation of any $\tilde{q}_i$ is $\kappa G_0 + (1-\kappa)G_{00}$ which further reduces to G_{00} if $G_0 = G_{00}$. Note that $(\tilde{q}_1, \dots, \tilde{q}_I) \sim semiHDP(\alpha, \gamma, \kappa, G_0, G_{00})$ defines an exchangeable prior over a vector of random probability measures.

We note several immediate yet interesting properties of the model. First, note that if $\kappa = 1$ in (6.6), then all the atoms and weights in the representation of the $\tilde{q}_i$'s are independent and different with probability one, since the beta distribution and G_0 are absolutely continuous. If $\kappa = 0$, then our prior (6.5)-(6.7) coincides with the Hierarchical Dirichlet Process in Teh et al. (2006). Since $\tilde{q}_0 = \sum_{h \geq 1} p_h \delta_{\tau_h}$, then, with positive probability, we have $\theta^*_{kh} = \theta^*_{k'm} = \tau_\ell$ for $k \neq k'$, i.e. all the $\tilde{q}_k$'s share the same atoms in the stick-breaking representation of $\tilde{q}_0$. However, even when $\kappa = 0$, $\tilde{q}_k \neq \tilde{q}_j$ with probability one, as the weights $\{w_{kh}\}_h$ and $\{w_{jh}\}_h$ are different, since they are built from (conditionally) independent stick-breaking priors. This is precisely the feature that allows us to circumvent the degeneracy problem.

Second, our model introduces a vector parameter $\boldsymbol{c}$, which assists selecting each population distribution from the finite set $\tilde{q}_1, \ldots, \tilde{q}_I$, in turn assumed to arise from the semi-HDP prior (6.5)-(6.7). The former allows two different populations to have the same distribution (or mixing measure) with positive probability, while the latter allows to overcome the degeneracy issue while retaining exchangeability. Indeed, as noted above, $\tilde{q}_i$ and $\tilde{q}_j$ may share atoms. The atoms in common arise from the atomicity of the base measure and we let the atomic component of the base measure to be a draw from a DP. The result is a very flexible model, that on one hand is particularly well-suited for problems such as density estimation, and on the other, can be used to construct clusters of the I populations, as desired.

6.2.3 A restaurant representation

To better understand the cluster allocation under model (6.3)-(6.7), we rewrite (6.3) introducing the latent variables $\{\theta_{ij}\}$ as follows

$$y_{ij} \mid \tilde{q}_1, \ldots \tilde{q}_I, \boldsymbol{c}, \theta_{ij} \overset{\text{ind}}{\sim} k(\cdot \mid \theta_{ij}) \tag{6.9}$$

$$\theta_{i1}, \ldots \theta_{iN_i} \mid \tilde{q}_1, \ldots \tilde{q}_I, \boldsymbol{c} \overset{\text{iid}}{\sim} \tilde{q}_{c_i} \tag{6.10}$$

and $\{\theta_{i\ell}\}_\ell \perp \{\theta_{jm}\}_m$ for $i \neq j$, conditionally on $\tilde{q}_1, \ldots, \tilde{q}_I$.

We first derive the conditional law of the θ_{ij}'s under (6.9) - (6.10), and (6.4)-(6.6), given $\tilde{q}_0$. All customers of group i enter restaurant r (such that $c_i = r$). If group i is the first group entering restaurant r, then the usual Chinese Restaurant metaphor applies. Instead, let us imagine that group i is the last group entering restaurant r among those such that $c_m = r$. Upon entering the restaurant, the customer is presented with the usual Chinese Restaurant Process (CRP), so that

$$\theta_{ij} \mid \boldsymbol{c}, \{\theta_{mk},\ \forall m : c_m = c_i = r\}, \theta_{i1}, \ldots, \theta_{ij-1}, \tilde{q}_0 \sim \sum_{\ell=1}^{H_r} \frac{n_{r\ell}}{\alpha + n_{r\cdot}} \delta_{\theta^*_{r\ell}} + \frac{\alpha}{\alpha + n_{r\cdot}} \tilde{p}, \tag{6.11}$$

that is the CRP when considering all the groups entering restaurant r as a single group. Here H_r denotes the number of tables in restaurant r, and $n_{r\ell}$ is the number of customers who entered from restaurant r and are seating at table ℓ. Moreover, note that $\theta^*_{r\ell} \mid \tilde{q}_0 \overset{\text{iid}}{\sim} \tilde{p}$, so that, as in the HDP, there might be ties among the $\theta^*_{r\ell}$ also when keeping r fixed. This is an important observation as the fact that there might be ties for different values of $r \neq r'$ instead, is exactly what let us avoid the degeneracy to the exchangeable case. Note that (6.11) holds also for θ_{i1}, i.e. the first customer in group i. In the following, we will use *clusters* or *tables* interchangeably. However, note that, unlike traditional CRPs, the number of *clusters* does not coincide with the number of unique values in a sample. This point is clarified in Argiento et al. (2019), who introduce the notion of ℓ–cluster, which is essentially the *table* in our restaurant metaphor.

Observe from (6.11) that when a new cluster is created, its label is sampled from $\tilde{p}$. In practice, we augment the parameter space with a new binary latent variable for each cluster, namely $h_{r\ell}$, with $h_{r\ell} \overset{\text{iid}}{\sim} \text{Bernoulli}(\kappa)$, so that

$$\theta^*_{r\ell} \mid h_{r\ell} = 1 \ \sim G_0 \qquad \text{and} \qquad \theta^*_{r\ell} \mid h_{r\ell} = 0, \tilde{q}_0 \ \sim \tilde{q}_0.$$

Upon conditioning on $\{h_{r\ell}\}$ it is straightforward to integrate out $\tilde{q}_0$. Indeed, we can write

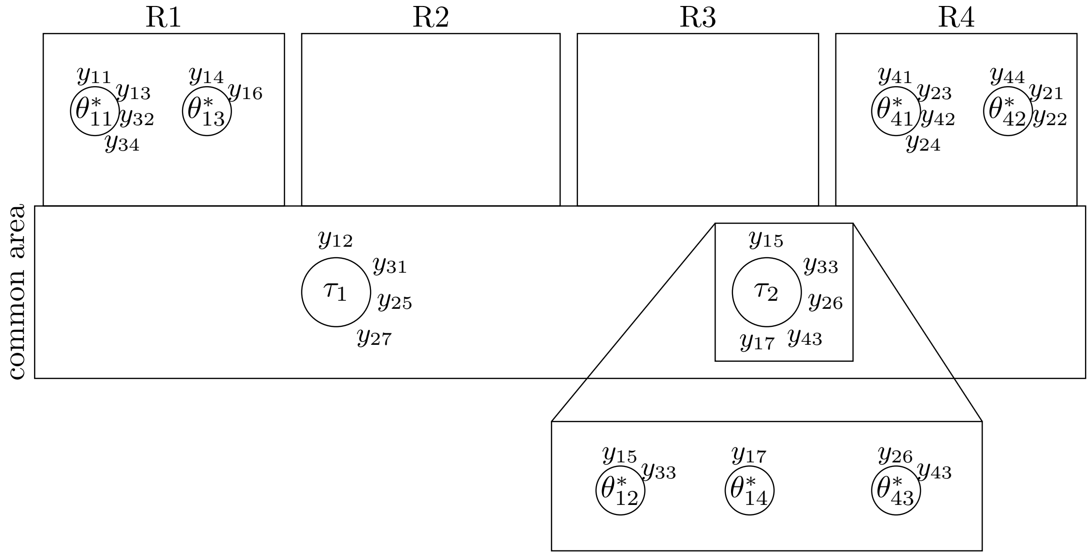


Figure 6.2.1: Restaurant representation of the semi-HDP allocation. In the image, $\boldsymbol{c} = (1, 4, 1, 4)$ so that groups one and three enter in restaurant R1 while groups two and four enter in restaurant R4. In the 'common area' two tables are represented, τ_1 and τ_2. 'Zooming' into τ_2 shows that there are three different θ^*'s associated to the value τ_2, namely $\theta^*_{12}, \theta^*_{13}$ and θ^*_{43}. The first two originate from R1, showing that it is possible to have ties among the θ^*'s even inside the same restaurant, while the table labeled θ^*_{43} shows that it is possible to have ties across different restaurants.

the joint distribution of $\{\theta^*_{r\ell},\ \forall r\ \forall \ell\}$, conditional on $\{h_{r\ell}\}$ as

$$\{\theta^*_{r\ell}\} \mid \{h_{r\ell}\}, \tilde{q}_0 \sim \prod_{r,\ell} G_0(d\theta^*_{r\ell})^{h_{r\ell}} \prod_{r,\ell} \tilde{q}_0(d\theta^*_{r\ell})^{1-h_{r\ell}}.$$

Hence we see that $\{\theta^*_{r\ell},\ \forall r\ \forall \ell : h_{r\ell} = 0\}$ is a conditionally i.i.d sample from $\tilde{q}_0$ (given all the h_{rl}'s and $\tilde{q}_0$), so that we can write:

$$\theta^*_{r\ell} \mid h_{r\ell} = 0, \{\theta^*_{ij} : h_{ij} = 0\} \sim \sum_{k=1}^{H_0} \frac{m_{\cdot k}}{m_{\cdot\cdot} + \gamma} \delta_{\tau_k} + \frac{\gamma}{m_{\cdot\cdot} + \gamma} G_{00} \tag{6.12}$$

and $\tau_k \overset{\text{iid}}{\sim} G_{00}$, where H_0 denotes the number of tables in the common area in Figure 6.2.1, and m_{rk} denotes the cardinality of the set $\{\theta^*_{r\ell} : \theta^*_{r\ell} = \tau_k\}$. The dot subindex denotes summation over the corresponding subindex values. Hence, conditioning on all the (r, ℓ) such that $h_{r\ell} = 0$, with r corresponding to a non-empty restaurant, we recover the Chinese Restaurant Franchise (CRF) that describes the HDP.

We can describe the previously discussed clustering structure in terms of a restaurant metaphor as the 'food court of Chinese restaurants with private and shared areas'. Here, the $\theta^*_{r\ell}$ correspond to the tables and θ_{ij} to the customers. Moreover, a dish is associated to each table. Dishes are represented by the various $\theta^*_{r\ell}$'s . There is one big common area where tables are shared among all the restaurants and I additional "private" small rooms, one per restaurant, as seen in Figure 6.2.1. The common area accommodates tables arising from the HDP, i.e. those tables such that $\tau_k \overset{\text{iid}}{\sim} G_{00}$, while the small rooms host those tables associated to non empty restaurants, such that $\theta^*_{r\ell} \mid h_{r\ell} = 1 \overset{\text{iid}}{\sim} G_0$. All the customers of group i enter restaurant r (such that $c_i = r$). Upon entering the restaurant, a customer is presented with a menu. The H_r dishes in the menu are the $\theta^*_{r\ell}$'s, and because $\theta^*_{r\ell} \overset{\text{iid}}{\sim} \tilde{p}$, there might be repeated dishes; see (6.11). The customer either chooses one of the dishes in the menu, with probability proportional to the number of customers who entered the same restaurant and chose that dish, or a new dish (that is not included in the menu yet) with probability proportional to α; again, see (6.11). If the latter option is chosen, with probability κ a new table is created in the restaurant-specific area, H_r is incremented by one and a new dish $\theta^*_{rH_r+1}$ is drawn from G_0. With probability $1-\kappa$ instead, the customer is directed to the shared area, where (s)he chooses to seat in one of the occupied tables with a probability proportional to $m_{\cdot k}$, i.e. the number of items in the menus (from all the restaurants) that are equal to dish τ_k, or seats at a new table with a probability proportional to γ, as seen from (6.12). We point out that the choice of table in this case is made without any knowledge of which restaurant the dishes came from. Moreover, if the customer chooses to sit at a new table, we increment H_0 by one and draw $\tau_{H_0+1} \sim G_{00}$; we also increment H_r by one and set $\theta^*_{rH_r+1} = \tau_{H_0+1}$. Observe that in the original CRF metaphor, it is not the tables that are shared across restaurants, but rather the dishes. In our metaphor instead, we group together all the tables corresponding to the same τ_h and place them in the shared area. This is somewhat reminiscent of the direct sampler scheme for the HDP. Nevertheless, observe that the bookkeeping of the m_{rk}'s is still needed. To exemplify this, in Figure 6.2.1 we report a 'zoom' on a particular shared table τ, showing that the θ^*'s associated to that table are still present in our metaphor, but can be collapsed into a single shared table when it is convenient.

6.3 Theoretical properties of the semi-HDP prior

Here we develop additional properties of the proposed prior model. In particular, we study the topological support of the semi-HDP and show how exactly the degeneracy issue is resolved by studying the induced joint random partition model on the I populations.

6.3.1 Support and moments

An essential requirement of nonparametric priors is that they should have large topological support; see Ferguson (1973). Let us denote by $\pi_{\tilde{\boldsymbol{p}}}$ the probability measure on $\mathbb{P}^I_\Theta$ corresponding to the prior distribution $\pi(\tilde{p}_1, \ldots, \tilde{p}_I)$ of the random vector $(\tilde{p}_1, \ldots, \tilde{p}_I)$ specified in (6.4)–(6.7), with $\tilde{p}_i = \tilde{q}_{c_i}$; see (6.1). We show here that the prior probability measure $\pi_{\tilde{\boldsymbol{p}}}$ has full weak support, i.e. given any point $\boldsymbol{g} = (g_1, \ldots, g_I)$ in $\mathbb{P}^I_\Theta$, $\pi_{\tilde{\boldsymbol{p}}}$ gives positive mass to any weak neighborhood $\mathcal{U}(\boldsymbol{g}; \epsilon)$ of $\boldsymbol{g}$, of diameter ϵ.

Proposition 6.1 (Full Weak Support). *Let $\pi_{\tilde{\boldsymbol{p}}}(g_1 \ldots, g_I)$ be the prior probability measure on $\mathbb{P}^I_\Theta$ defined by* (6.4)*–*(6.7)*.*

(a) *If G_0 in* (6.6) *has full support on Θ and $0 < \kappa \leq 1$, then $\pi_{\tilde{\boldsymbol{p}}}(g_1 \ldots, g_I)$ has full weak support.*

(b) *If $\kappa = 0$ and G_{00} in* (6.7) *has full support, then $\pi_{\tilde{\boldsymbol{p}}}(g_1 \ldots, g_I)$ has full weak support.*

It is straightforward to show that in case where $\pi_c(c_1, \ldots, c_I)$ is exchangeable and $P(c_i = \ell) = \omega_\ell$ for $\ell = 1, \ldots, I$ then (6.3)–(6.7) becomes, after marginalizing with respect to $\boldsymbol{c}$,

$$y_{ij} \mid \tilde{q}_1, \ldots, \tilde{q}_I \overset{\text{iid}}{\sim} \sum_{\boldsymbol{c}} \int_\Theta k(\cdot \mid \theta)\, \tilde{q}_{c_i}(d\theta) \pi_c(c_1, \ldots, c_I) = \sum_{\ell=1}^I \omega_\ell \int_\Theta k(\cdot \mid \theta)\, \tilde{q}_\ell(d\theta).$$

In this case, the conditional marginal distribution of each observation can be expressed as a finite mixture of mixtures of the density $k(\cdot \mid \theta)$ with respect to each of the random measures $\tilde{q}_1, \dots, \tilde{q}_I$, i.e. a finite mixture of Bayesian nonparametric mixtures.

We have mentioned above that in the case in which $G_{00} = G_0$ in Equations (6.6) - (6.7), the marginal law of $\tilde{q}_i$ is G_0, and equivalently, for each $A \in \mathcal{B}(\Theta)$, $\mathbb{E}[\tilde{q}_i(A)] = G_0(A)$ for any i. In this case, the covariance between $\tilde{q}_1$ and $\tilde{q}_2$ is given by

$$\operatorname{cov}\left(\tilde{q}_1(A), \tilde{q}_2(B)\right) = \frac{(1-\kappa)^2}{1+\gamma}\left(G_0(A \cap B) - G_0(A)G_0(B)\right).$$

See Appendix 6.A, for the proof of these formulas. Note that, in the case of Hierarchical Normalized Completely Random Measures, and hence in the HDP, the covariance between $\tilde{q}_1$ and $\tilde{q}_2$ depends exclusively on the intensity of the random measure governing $\tilde{q}_0$ (in the case of the DP the dependence is on γ). For instance, see Argiento et al. (2019), Equation (5) in the Supplementary Material. Instead, in the Semi-HDP, an additional parameter can be used to tune such covariance: the weight κ. Indeed, as κ approaches 1, the two measures become more and more uncorrelated, the limiting case being full independence as discussed at the end of Section 6.2.2. In Appendix 6.A, we also report an expression for the higher moments of $\tilde{q}_i(A)$ for any i.

6.3.2 Degeneracy and marginal law

We now formalize the intuition given in Section 6.2.3 and show that our model, as defined in (6.3)-(6.7), does not incur in the degeneracy issue described by Camerlenghi et al. (2019). The degeneracy of a nested nonparametric model refers to the following situation: if there are shared values (or atoms in the corresponding mixture model) across any two populations, then the posterior of these population/random probabilities degenerates, forcing homogeneity across the corresponding samples. See also the discussion in Beraha and Guglielmi (2019).

From the food court metaphor described above, it is straightforward to see that degeneracy is avoided if two customers sit in the same table (of the common area) with positive probability, conditioning on the event that they entered from two different restaurants.

To see that this is so for the proposed model, let us consider the case $I = 2$ and $\theta_{i1} \mid \tilde{q}_1, \tilde{q}_2, \boldsymbol{c} = (1,2) \sim \tilde{q}_i$, for $i = 1, 2$. Marginalizing out $(\tilde{q}_1, \tilde{q}_2)$, this is equivalent to $\theta_{11}, \theta_{21} \mid \tilde{q}_0,\ \boldsymbol{c} = (1,2) \overset{\text{iid}}{\sim} wG_0 + (1-w)\tilde{q}_0$. Now, since G_0 is absolutely continuous, $\{\theta_{11} = \theta_{21}\}$ if and only if (i) θ_{11} and θ_{21} are sampled i.i.d. from $\tilde{q}_0$; and (ii) we have a tie (which arises from the Pólya-urn scheme), i.e. $\theta_{21} = \tau_1 = \theta_{11}$ and $\tau_1 \sim G_{00}$. This means that θ_{11}, the first customer, sits in a table of the common area, an event that happens with probability $1 - \kappa$ since she is the first one in the whole system, and θ_{21} decides to sit in the common area (with probability $1 - \kappa$) and subsequently decides to sit at the same table of θ_{11} (which happens with probability $\frac{1}{\gamma+1}$). Summing up we have that $p(\theta_{11} = \theta_{21} \mid \boldsymbol{c} = (1,2)) = (1-\kappa)^2/(1+\gamma)$ which is strictly positive if $\kappa < 1$. Hence, by Bayes' rule, we have that

$$P(c_1 \neq c_2 \mid \theta_{11} = \theta_{21}) = \frac{P(\theta_{11} = \theta_{21} \mid c_1 \neq c_2) P(c_1 \neq c_2)}{\sum_{i,j} P(\theta_{11} = \theta_{21} \mid \boldsymbol{c} = (i,j)) P(\boldsymbol{c} = (i,j))} > 0.$$

Moreover, when $\kappa = 1$ we find the same degeneracy issue described in Camerlenghi et al. (2019), as proved in Proposition 6.2 below.

To get a more in-depth look at these issues, we follow Camerlenghi et al. (2019) and study properties of the partially exchangeable partition probability function (pEPPF) induced by our model, which we define in the special case of $I = 2$. Consider a sample $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ of size $N = N_1 + N_2$ from model (6.10), together with (6.4)-(6.7) for $I = 2$

populations; let $k = k_1 + k_2 + k_0$ the number of unique values in the samples, with k_1 (k_2) unique values specific to group 1 (2) and k_0 shared between the groups. Call $\boldsymbol{n}_i$ the frequencies of the k_i unique values in group i and $\boldsymbol{q}_i$ the frequencies of the k_0 shared values in group i; this is the same notation as in Camerlenghi et al. (2019), Section 2.2. The pEPPF is defined as

$$\Pi_k^N(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1, \boldsymbol{q}_2 \mid \boldsymbol{c} = (\ell, m)) =$$
$$\int_{\Theta^k} \mathbb{E}\left[\prod_{j=1}^{k_1} \tilde{q}_\ell^{n_{1j}}(d\theta_{1j}^*) \prod_{j=1}^{k_2} \tilde{q}_m^{n_{2j}}(d\theta_{2j}^*) \prod_{j=1}^{k_0} \tilde{q}_\ell^{q_{1j}}(d\tau_j)\tilde{q}_m^{q_{2j}}(d\tau_j)\right]$$

Proposition 6.2. *Let κ in* (6.6) *be equal to 1, let $\pi_1 = P(c_1 = c_2)$, then the pEPPF $\Pi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1, \boldsymbol{q}_2)$ can be expressed as:*

$$\begin{aligned}\Pi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1, \boldsymbol{q}_2) &= \pi_1 \Phi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1 + \boldsymbol{q}_2) \\ &\quad + (1 - \pi_1)\Phi_{k_0+k_1}^{(N_1)}(\boldsymbol{n}_1, \boldsymbol{q}_1)\Phi_{k_0+k_1}^{(N_2)}(\boldsymbol{n}_2, \boldsymbol{q}_2) I(k_0 = 0) \qquad (6.13)\end{aligned}$$

where

$$\Phi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1 + \boldsymbol{q}_2) = \frac{\alpha^{k_1+k_2+k_0}\Gamma(\alpha)}{\Gamma(\alpha + N)} \prod_{j=1}^{k_1} \Gamma(n_{1j}) \prod_{j=1}^{k_2} \Gamma(n_{2j}) \prod_{j=1}^{k_0} \Gamma(q_{1j} + q_{2j})$$

is the EPPF of the fully exchangeable case, and

$$\Phi_{k_0+k_i}^{(N_i)}(\boldsymbol{n}_i, \boldsymbol{q}_i) = \frac{\alpha^{k_i+k_0}\Gamma(\alpha)}{\Gamma(\alpha + N_i)} \prod_{j=1}^{k_i} \Gamma(n_{ij}) \prod_{j=1}^{k_0} \Gamma(q_{ij}), \ i = 1, 2$$

is the marginal EPPF for the individual group i.

This result shows that a suitable prior for κ requires assigning zero probability to the event $\kappa = 1$. The assumption in (6.8) trivially satisfies this requirement.

Finally, we consider the marginal law of a sequence of vectors $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_I)$, $\boldsymbol{\theta}_\ell = (\theta_{\ell 1}, \ldots \theta_{\ell N_l})$ from model (6.3)-(6.7). Let us first derive the marginal law conditioning on $\boldsymbol{c}$, as the full marginal law will be the mixture of these conditional laws over all the possible values of $\boldsymbol{c}$.

Proposition 6.3. *The marginal law of a sequence of vectors $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_I)$, $\boldsymbol{\theta}_\ell = (\theta_{\ell 1}, \ldots \theta_{\ell N_\ell})$ from model* (6.3)-(6.7)*, conditional to $\boldsymbol{c}$ is*

$$\prod_{i=1}^{R(\boldsymbol{c})} eppf(\boldsymbol{n}_{r_i}; \alpha) \sum_{\boldsymbol{h} \in \{0,1\}^L} p(\boldsymbol{h}) \prod_{\ell=1}^{L} G_0(d\theta_\ell^*)^{h_\ell} \times eppf(\boldsymbol{m}_{r_i} \mid \boldsymbol{h}; \gamma) \prod_{k=1}^{M} G_{00}(d\theta_k^{**}). \qquad (6.14)$$

Here, $\{\theta_\ell^*\}_{\ell=1}^L = \{\theta_{11}^*, \ldots, \theta_{IH_I}^*\}$ is a sequence representing all the tables in the process, obtained by concatenating the tables in each restaurant. Moreover, $R(\boldsymbol{c})$ is the number of unique values in $\boldsymbol{c}$, i.e. the number of non-empty restaurants, $\boldsymbol{n}_{r_i}$ is the vector of ℓ-cluster sizes for restaurant r_i, $\boldsymbol{m}_{r_i}$ is the vector of the cluster sizes of the θ_ℓ^* such that $h_\ell = 0$ and θ_k^{**} are the unique values among such θ_ℓ^*, where 'eppf' denotes the the distribution of the partition induced by the table assignment procedure in the food court of Chinese restaurants described in Section 6.2.3.

The marginal law of $(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_I)$ is then

$$\mathcal{L}(d\boldsymbol{\theta}_1, \dots, d\boldsymbol{\theta}_I) = \sum_{\boldsymbol{c}} \mathcal{L}(d\boldsymbol{\theta}_1, \dots, d\boldsymbol{\theta}_I \mid \boldsymbol{c}) \pi(\boldsymbol{c})$$

where $\mathcal{L}(d\boldsymbol{\theta}_1, \dots, d\boldsymbol{\theta}_I \mid \boldsymbol{c})$ is given in (6.14).

Observe that in Proposition 6.2 we denoted by Φ the EPPF, while in (6.14) we use notation '*eppf*'. This is to remark that these objects are inherently different: Φ is the EPPF of the partition of *unique* values in the sample, while *eppf* here is the EPPF of the tables, or ℓ–clusters, induced by the table assignment procedure described in Section 6.2.3. Hence, from a sample $\boldsymbol{\theta}$ one can recover $\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1, \boldsymbol{q}_2$ in (6.13) but not $\boldsymbol{n}_{r_i}$ in (6.14).

6.3.3 Some results on the Bayes factor for testing homogeneity

We consider now testing for homogeneity within the proposed partial exchangeability framework. As a byproduct of the assumed model, the corresponding Bayes factor is immediately available. For example, if one wanted to test whether populations i and j were homogeneous, it would suffice to compute the Bayes factor for the test

$$H_0 : c_i = c_j \quad \text{vs.} \quad H_1 : c_i \neq c_j \tag{6.15}$$

which can be straightforwardly estimated from the output of the posterior simulation algorithm that will be presented later on. Note that these 'pairwise' homogeneity tests are not the only object of interest that we can tackle within our framework. Indeed it is possible to test any possible combination of $\boldsymbol{c}$ against an alternative.

These tests admit an equivalent representation in terms of a model selection problem; for example in the case of $I = 2$ populations, we can rewrite (6.15), for $i = 1$ and $j = 2$, as a model selection test for M_1 against M_2, where

$$M_1 : \; y_{11}, \dots, y_{1N_1}, y_{21}, \dots, y_{2N_2} \mid \tilde{q}_1 \overset{\text{iid}}{\sim} \int_\Theta k(\cdot \mid \theta) \tilde{q}_1(d\theta)$$
$$\tilde{q}_1 \sim \mathit{semiHDP}(\alpha, \gamma, \kappa, G_0, G_{00})$$

and

$$M_2 : \; y_{i1}, \dots, y_{iN_i}, \mid \tilde{q}_i \overset{\text{iid}}{\sim} \int_\Theta k(\cdot \mid \theta) \tilde{q}_i(d\theta), \; i = 1, 2$$
$$\tilde{q}_1, \tilde{q}_2 \sim \mathit{semiHDP}(\alpha, \gamma, \kappa, G_0, G_{00}).$$

In this case

$$BF_{12} := BF_{12}(y_{11}, \dots, y_{1N_1}, y_{21}, \dots, y_{2N_2}) = \frac{m_{M_1}(y_{11}, \dots, y_{1N_1}, y_{21}, \dots, y_{2N_2})}{m_{M_2}(y_{11}, \dots, y_{1N_1}, y_{21}, \dots, y_{2N_2})},$$

where m_{M_i} denotes the marginal law of the data under model M_i, $i = 1, 2$, defined above. Asymptotic properties of Bayes factors have been discussed by several authors. We refer to Walker et al. (2004), Ghosal et al. (2008) for a more detailed discussion and to Chib and Kuffner (2016) for a recent survey on the topic. Chatterjee et al. (2020) is a recent and solid contribution to the almost sure convergence of Bayes factor in the general set-up that includes dependent data, i.e. beyond the usual i.i.d. context.

In words, our approach can be described as follows. When the data are assumed to be exchangeable, we assume that both samples are generated i.i.d from a distribution P_0 with density p_0. If the data are instead assumed to be partially exchangeable, then we consider the first population to be generated i.i.d from a certain P_0 with density p_0, while the second one is generated from Q_0 with density q_0, with $P_0 \neq Q_0$ and independence holds across populations. The Bayes factor for comparing M_1 against M_2 is thus consistent if: (i) $BF_{12} \to +\infty$ P_0^∞–a.s. when $N_1, N_2 \to +\infty$ if the groups are truly homogeneous, and (ii) $BF_{12} \to 0$ $(P_0 \otimes Q_0)^\infty$–a.s. when $N_1, N_2 \to +\infty$ if the groups are not homogeneous.

The two scenarios must be checked separately. In the latter case, consistency of the Bayes factor can be proved by arguing that only model M_2 satisfies the so-called Kullback-Leibler property, so that consistency is ensured by the theory in Walker et al. (2004). We summarize this result in the following proposition.

Proposition 6.4. *Assume that $y_{11}, \ldots, y_{1N_1} \overset{iid}{\sim} P_0$, $y_{21}, \ldots, y_{2N_2} \overset{iid}{\sim} Q_0$, $P_0 \neq Q_0$, and that $\{y_{1i}\}$ and $\{y_{2j}\}$ are independent. Assume that P_0 and Q_0 are absolutely continuous measures with probability density functions p_0 and q_0 respectively. Then, under conditions $B1$-$B9$ in Wu and Ghosal (2008), $BF_{12} \to 0$ as $N_1, N_2 \to +\infty$.*

Observe that, out of the nine conditions $B1$-$B9$, we have that $B1 - B3$, $B7$ and $B9$ involve regularity conditions of the kernel $k(\cdot|\theta)$. These are satisfied if the kernel is, for example, univariate Gaussian with parameters $\theta = (\mu, \sigma^2)$. Conditions $B4 - B6$ involve regularity of the true data generating density, which are usually satisfied in practice. Condition $B8$ requires that the mixing measure has full weak support, already proved in Proposition 6.1.

On the other hand, when $p_0 = q_0$, consistency of the Bayes factor would require $BF_{12} \to +\infty$. This is a result we have not been able to prove so far. Appendix 6.B discusses the relevant issues arising when trying to prove the consistency in this setting; we just report here that the key missing condition is an upper bound of the prior mass of M_2. The lack of such bounds for general nonparametric models is well known in the literature, and not specific to our case, as it is shared, for instance, by Bhattacharya and Dunson (2012) and Tokdar and Martin (2019). In both cases, the authors were able to prove the consistency under the alternative hypothesis but not under the null. For a discussion on the 'necessity' of these bounds in nonparametric models, see Tokdar and Martin (2019).

In light of the previous consistency result for the non-homogeneous case, our recommendation to carry out the homogeneity test is to decide in favor of H_0 whenever the posterior of c_i, c_j does not strongly concentrate on $c_i \neq c_j$. As Section 6.5 shows, in our simulated data experiments this choice consistently identifies the right structure of homogeneity among populations. See also the discussion later in Section 6.7.

6.4 Posterior Simulation

We illustrate an MCMC sampler based on the restaurant representation derived in Section 6.2.3. The random measures $\{\tilde{q}_i\}_i$ and $\tilde{q}_0$ are marginalized out for all the updates except for the case of $\boldsymbol{c}$, for which we use a result from Pitman (1996) to sample from the full conditional of each $\tilde{q}_i$, truncating the infinite stick-breaking sum adaptively; see below. We refer to this algorithm as *marginal*. We also note that, by a prior truncation of all the stick-breaking infinite sums to a fixed number of atoms, we can derive a blocked Gibbs sampler as in Ishwaran and James (2001). However, in our applications the blocked Gibbs sampler was significantly slower both in reaching convergence to the stationary distribution and to complete one single iteration of the MCMC update. Hence, we will describe and use only the marginal algorithm.

We follow the notation introduced in Section 6.2.3. The state of our MCMC sampler consists of the restaurant tables $\{\theta^*_{rh}\}$, the tables in the common area $\{\tau_h\}$, a set of binary variables $\{h_{rj}\}$, indicating if each table is 'located' in the restaurant-specific or in the common area, a set of discrete shared table allocation variables $t_{r\ell}$, one for each $\theta^*_{r\ell}$ such that $\theta^*_{r\ell} = \tau_k$ iff $t_{r\ell} = k$ and $h_{r\ell} = 0$, the categorical variables c_i, indicating the restaurant for each population, $\kappa \in (0, 1)$, and the table allocation variable s_{ij}: for each observation such that $\theta_{ij} = \theta^*_{rh}$ iff $c_i = r$ and $s_{ij} = h$. We also denote by H_0 and H_r the number of tables occupied in the shared area and in restaurant r respectively, m_{rk} indicates the number of customers in the common area entered from restaurant r seating at table k.

We use the dot notation for marginal counts, for example $n_{r\cdot}$ indicates all the customers entered in restaurant r. We summarize the Gibbs sampling scheme next.

- Sample the cluster allocation variables using the Chinese Restaurant Process,

  $$p(s_{ij} = s \mid c_i = r, rest) \propto \begin{cases} n^{-ij}_{r\ell} k(y_{ij} \mid \theta^*_{rh}) & \text{if } s \text{ previously used} \\ \alpha p(y_{ij} \mid \boldsymbol{s}^{-ij}, rest) & \text{if } s = s^{new} \end{cases} \tag{6.16}$$

  where

  $$\begin{aligned} p(y_{ij} \mid \boldsymbol{s}^{-ij}, rest) &= \kappa \int k(y_{ij} \mid \theta) G_0(d\theta) + \\ &+ (1-\kappa) \left(\sum_{k=1}^{H_0} \frac{m^{-ij}_{\cdot k}}{m^{-ij}_{\cdot\cdot} + \gamma} k(y_{ij} \mid \tau_k) + \frac{\gamma}{m^{-ij}_{\cdot\cdot} + \gamma} \int k(y_{ij} \mid \theta) G_{00}(d\theta) \right), \end{aligned} \tag{6.17}$$

  and where the notation x^{-ij} means that observation y_{ij} is removed from the calculations involving the variable x.

  If $s = s^{new}$, a new table is created. The associated value $\theta^*_{rs^{new}}$ is sampled from G_0 with probability κ or from $\tilde{q}_0$ with probability $1-\kappa$, as described in Section 6.2.3. The corresponding latent variables $h_{rs^{new}}$ and $t_{rs^{new}}$ are set accordingly. When sampling from (6.12) a new table in the shared area might be created. In that case, $t_{rs^{new}}$ is set to $H_0 + 1$.

- Sample the table allocation variables $t_{r\ell}$ as in the HDP:

  $$p(t_{r\ell} = k \mid rest) \propto \begin{cases} m^{-r\ell}_{\cdot k} \displaystyle\prod_{(i,j): c_i = r, s_{i,j} = \ell} k(y_{ij} \mid \tau_k) & \text{if } k \text{ previously used} \\ \gamma \displaystyle\int \prod_{(i,j): c_i = r, s_{i,j} = \ell} k(y_{ij} \mid \tau) G_{00}(d\tau) & \text{if } k = k^{new}, \end{cases} \tag{6.18}$$

  where the notation $x^{-r\ell}$ means that table $\theta^*_{r\ell}$, including all the associated observations, is entirely removed from the calculations involving variable x. If $k = k^{new}$ a new table is created in the shared area, the allocation variables s_{ij} are left unchanged.

- Sample the cluster values from

  $$\mathcal{L}(\theta^*_{r\ell} \mid h_{r\ell} = 1, rest) \propto G_0(\theta^*_{r\ell}) \prod_{(i,j): c_i = r, s_{i,j} = \ell} k(y_{ij} \mid \theta^*_{r\ell})$$

  and

  $$\mathcal{L}(\tau_k \mid rest) \propto G_{00}(\tau_k) \prod_{(i,j) \in (*)} k(y_{ij} \mid \tau_k)$$

  where the product $(*)$ is over all the index couples such that $c_i = r, s_{ij} = \ell$, $h_{r\ell} = 0$ and $\theta^*_{r\ell} = \tau_k$. Observe that, when $h_{r\ell} = 0$, it means that $\theta^*_{r\ell} = \tau_k$ for some k. Hence, in this case, $\theta^*_{r\ell}$ is purely symbolic and we do not need to sample a value for it.

- Sample each $h_{r\ell}$ independently from

$$p(h_{r\ell} = 1 | rest) \propto \kappa G_0(\theta^*_{r\ell})$$
$$p(h_{r\ell} = 0 | rest) \propto (1-\kappa) \left(\sum_{k=1}^{H_0} \frac{m_{\cdot k}^{-r\ell}}{m_{\cdot\cdot}^{-r\ell} + \gamma} \delta_{\tau_k}(\theta^*_{r\ell}) + \frac{\gamma}{m_{\cdot\cdot}^{-r\ell} + \gamma} G_{00}(\theta^*_{r\ell}) \right),$$

  where, as in (6.18), the notation $x^{-r\ell}$ means that table $\theta^*_{r\ell}$, including all its associated observations, is removed from the calculations involving variable x. Observe that, while in the update of the cluster values all the $\theta^*_{r\ell}$ referring to the same τ_k were updated at once, here we move the tables one by one.

- Sample κ from $\mathcal{L}(\kappa \,|\, rest) \sim Beta\left(a_\kappa + \sum_{i,j} h_{ij},\ b_\kappa + \sum_{i,j}(1 - h_{ij})\right)$.

- Sample $\boldsymbol{\omega}$ from

$$\boldsymbol{\omega} \,|\, rest \sim Dirichlet\Big(\eta_1 + \sum_{i=1}^{I} \mathbb{I}[c_i = 1], \ldots, \eta_I + \sum_{i=1}^{I} \mathbb{I}[c_i = I]\Big)$$

  where $\mathbb{I}[\cdot]$ denotes the indicator function.

- Sample each c_i in $\boldsymbol{c} = (c_1, \ldots, c_I)$ independently from

$$P(c_i = r \,|\, \tilde{q}_1, \ldots, \tilde{q}_I, \boldsymbol{\omega}, \boldsymbol{y}_i) \propto \omega_r \prod_{j=1}^{N_i} \int k(y_{ij} \,|\, \theta) \tilde{q}_r(d\theta). \tag{6.19}$$

  If the new value of c_i differs from the previous one, then following (6.16), all the observations $y_{i1}, \ldots, y_{iN_i}$ are reallocated to the new restaurant.

Note that the update in (6.19) involves the previously marginalized random probability measures $\tilde{q}_1, \ldots, \tilde{q}_I$. Thus, before performing this update, we need to draw the $\tilde{q}_i$'s from their corresponding full conditional distributions. It follows from Corollary 20 in Pitman (1996) that the conditional distribution of $\tilde{q}_r$ given $\boldsymbol{c}$, $\boldsymbol{n}_r$, $\boldsymbol{\theta}^*_r$, κ, and $\tilde{q}_0$ coincides with the distribution of $\pi_{r0}\tilde{q}'_r + \sum_{h=1}^{H_r} \pi_{rh}\delta_{\theta^*_{rh}}$, where $(\pi_{r0}, \pi_{r1}, \ldots, \pi_{rH_r}) \sim Dirichlet(\alpha, n_{r1}, \ldots, n_{rH_r})$ and $\tilde{q}'_r \,|\, \tilde{q}_0 \sim DP(\alpha, \tilde{p})$. This result was employed in Taddy et al. (2012) to quantify posterior uncertainty of functionals of a Dirichlet process, and also in Canale et al. (2019) to derive an alternative MCMC scheme for mixture models. It follows from the usual stick breaking representation that $\tilde{q}'_r = \sum_{h=1}^{\infty} w'_{rh}\delta_{\theta'_{rh}}$ with $\{w'_{rh}\}_h \sim SB(\alpha)$ and $\theta'_{rh} \,|\, \kappa, \tilde{q}_0 \overset{\text{iid}}{\sim} \kappa G_0 + (1-\kappa)\tilde{q}_0$. Similarly, the conditional distribution of $\tilde{q}_0$ given $\boldsymbol{\tau}$ and $\boldsymbol{m}$ coincides with the distribution of $v_0\tilde{q}'_0 + \sum_{k=1}^{H_0} v_k\delta_{\tau_k}$, where $(v_0, v_1, \ldots, v_{H_0}) \sim Dirichlet(\gamma, m_{\cdot 1}, \ldots, m_{\cdot H_0})$ and $\tilde{q}'_0 \sim DP(\gamma, G_{00})$.

In practice, we draw each $\tilde{q}'_r$ by truncating the infinite sum. Note that we do not need to set a priori the truncation level. Instead, we can specify an upper bound for the error introduced by the truncation and set the level adaptively. In fact, as a straightforward consequence of Theorem 1 in Ishwaran and James (2002) we have that the total variation distance between $\tilde{q}'_r$ and its approximation with M atoms, say $\tilde{q}_r^{M\prime}$, is bounded by $\varepsilon_M = 1 - \sum_{h=1}^{M} w'_{rh}$ (see also Theorem 2 in Lijoi et al., 2020c). The error induced on $\tilde{q}_r$ is then bounded by $\pi_{r0}\varepsilon_M$. Note that simulation of the atoms θ'_{rh} involves the discrete measure $\tilde{q}'_0$. However, we only need to draw a finite number of samples from it, and not its full trajectory, so that no truncation is necessary for $\tilde{q}'_0$. For ease of bookkeeping, we employ retrospective sampling (Papaspiliopoulos and Roberts, 2008) to simulate the atoms. Alternatively, the classical CRP representation can be used. In our experiments, because $\sum_{h=1}^{H_r} n_{rh} \gg \alpha$ we have $\pi_{r0} \ll \sum_{h=1}^{H_r} \pi_{rh} \approx 1$. Thus, choosing a truncation level $M = 10$ always produces an error on $\tilde{q}_i$ lower than 10^{-4} (henceforth fixed as the truncation error threshold). Furthermore, we are often not even required to draw samples from $\tilde{q}_0'$.

Of the aforementioned steps, the bottleneck is the update of $\boldsymbol{c}$ because for each c_i we are required to evaluate the densities of N_i points in I mixtures. If $N_i = N$ for all i, the computational cost of this step is $O(NI^2)$, which can be extremely demanding for large values of I. We can mitigate the computational burden by replacing this Gibbs step with a Metropolis-within-Gibbs step, in the same spirit of the *Metropolised Carlin and Chib* algorithm proposed in Dellaportas et al. (2002). At each step we propose a move from $c_i^{(\ell)} = r$ to $c_i^{(\ell+1)} = m$ with a certain probability $p_i(m \mid r)$. The transition is then accepted with the usual Metropolis-Hastings rule, i.e. the new update becomes:

- Propose a candidate m by sampling $p_i(m \mid r)$

- Accept the move with probability q, where

$$q = \min \left[1, \frac{P(c_i = m) \prod_{j=1}^{N_i} \int k(y_{ij} \mid \theta) \tilde{q}_m(d\theta)}{P(c_i = r) \prod_{j=1}^{N_i} \int k(y_{ij} \mid \theta) \tilde{q}_r(d\theta)} \frac{p_i(r \mid m)}{p_i(m \mid r)} \right]$$

We call this alternative sampling scheme the Metropolised sampler. The key point is that if evaluating the proposal $p_i(\cdot|\cdot)$ has a negligible cost, the computational cost of this step will be $O(2NI)$ as for each data point we need to evaluate only two mixtures: the one corresponding to the current state $\tilde{q}_r$ and the one corresponding to the proposed state $\tilde{q}_m$. Of course, the efficiency and mixing of the Markov chain will depend on a suitable choice of the transition probabilities $p_i(\cdot|\cdot)$; some possible alternatives are discussed in Section 6.5.

When, at the end of an iteration, a cluster is left unallocated (or empty), the probability of assigning an observation to that cluster will be zero for all subsequent steps. As in standard literature, we employ a relabeling step that gets rid of all the unused clusters. However, this relabeling step is slightly more complicated since there are two different types of clusters: one arising from G_0 and ones arising from $\tilde{q}_0$. Details of the relabeling procedure are discussed in the Appendix 6.C.

6.4.1 Use of pseudopriors

The above mentioned sampling scheme presents a major issue that could severely impact the mixing. Consider as an example the case when $I = 2$; if, at iteration k, the state jumps to $c_1 = c_2 = 1$, then all the tables of the second restaurant would be erased from the state, because no observation is assigned to them anymore. Switching back to $c_1 \neq c_2$ would then require that the approximation of $\tilde{q}_2$ sampled from its prior distribution gives sufficiently high likelihood to either $\boldsymbol{y}_1$ or $\boldsymbol{y}_2$, an extremely unlikely event in practice.

To overcome this issue, we make use of pseudopriors as in Carlin and Chib (1995), that is, whenever a random measure $\tilde{q}_r$ in $(\tilde{q}_1, \ldots, \tilde{q}_I)$ is not associated with any group, we sample the part of the state corresponding to that measure (the atoms $\{\theta^*_{r\ell}\}$ and number of customers $\{n_{r\ell}\}$ in each restaurant) from its pseudoprior. From the computational point of view, this is accomplished by running first a preliminary MCMC simulation where the c_i's are fixed as $c_i = i$, and collecting the samples. Then, in the actual MCMC simulation, whenever restaurant r is empty we change the state by choosing at random one of the previous samples obtained with fixed c_i's. Note that this use of pseudopriors does not alter the stationary distribution of the MCMC chain. Furthermore, the way pseudopriors are collected and sampled from is completely arbitrary, and our proposed solution works well in practice. Other valid options include approximations based on preliminary chain runs, as discussed in Carlin and Chib (1995).

	(μ_1, σ_1)	(μ_2, σ_2)	(μ_3, σ_3)	(μ_4, σ_4)	w_1	w_2
Scenario I	(0.0, 1.0)	(5.0, 1.0)	(0.0, 1.0)	(5.0, 1.0)	0.5	0.5
Scenario II	(5.0, 0.6)	(10.0, 0.6)	(5.0, 0.6)	(0.0, 0.6)	0.9	0.1
Scenario III	(0.0, 1.0)	(5.0, 1.0)	(0.0, 1.0)	(5.0, 1.0)	0.8	0.2

Table 6.5.1: Parameters of the simulated datasets

Section 6.5 below contains extensive simulation studies that show that the proposed model can be used to efficiently estimate densities for each population. We also tried the case of a large number of populations, e.g. $I = 100$ without any significant loss of performance.

6.5 Simulation Study

In this section we investigate the ability of our model to estimate dependent random densities. We fix the kernel $k(\cdot|\theta)$ in (6.2) to be the univariate Gaussian density with parameter $\theta = (\mu, \sigma^2)$ (mean and variance, respectively). Both base measures G_0 and G_{00} are chosen to be

$$\mathcal{N}(\mu \,|\, 0, 10\sigma^2) \times inv - gamma(\sigma^2 \,|\, 1, 1),$$

and unless otherwise stated, with hyperparameters α, γ fixed to 1, $a_\kappa = b_\kappa = 2$, and $\boldsymbol{\eta} = (1/I, \ldots, 1/I)$. Chains were run for $100,000$ iterations after discarding the first $10,000$ iterations as burn-in, keeping one every ten iterations, resulting in a final sample size of $10,000$ MCMC draws.

6.5.1 Two populations

We first focus on the special case of $I = 2$ populations. Consider generating data as follows

$$\begin{aligned} y_{1j} &\overset{\text{iid}}{\sim} w_1\mathcal{N}(\mu_1, \sigma_1) + (1 - w_1)\mathcal{N}(\mu_2, \sigma_2) \quad j = 1, \ldots N_1 \\ y_{2j} &\overset{\text{iid}}{\sim} w_2\mathcal{N}(\mu_3, \sigma_3) + (1 - w_2)\mathcal{N}(\mu_4, \sigma_4) \quad j = 1, \ldots N_2, \end{aligned} \tag{6.20}$$

that is each population is a mixture of two normal components. This is the same example considered in Camerlenghi et al. (2019). Table 6.5.1 summarizes the parameters used to generate the data. Note that these three scenarios cover either the full exchangeability case across both populations (Scenario I), as well as the partial exchangeability between the two populations (scenarios II and III). For each case, we simulated $N_1 = N_2 = 100$ observations for each group (independently).

Table 6.5.2 reports the posterior probabilities of the two population being identified as equal for the three scenarios. We can see that our model recovers the ground truth. Moreover Figure 6.D.2 in the Appendix file shows the density estimates, i.e. the posterior mean of the density evaluated on a fixed grid of points, together with pointwise 95% posterior credible intervals at each point x in the grid, obtained by our MCMC for scenarios I and III. Here, densities are estimated from the corresponding posterior mean evaluated on a fixed grid of points, while credible intervals are obtained by approximating the $\tilde{q}_i$'s as discussed in Section 6.4. We can see that in both the cases, locations and scales of the populations are recovered perfectly, while it seems that the weights of the mixture components are slightly more precise in Scenario I than in Scenario III.

Comparing the Bayes Factors shown in Table 6.5.2 with the ones in Camerlenghi et al. (2019) (5.86, 0.0 and 0.54 for the three scenarios, respectively), we see that both models are able to correctly assess homogeneity. However, the Bayes Factors obtained under our

	$P(c_1 = c_2 \mid data)$	BF_{01}
Scenario I	0.99	98.9
Scenario II	0.0	0.0
Scenario III	0.0	0.0

Table 6.5.2: Posterior inference

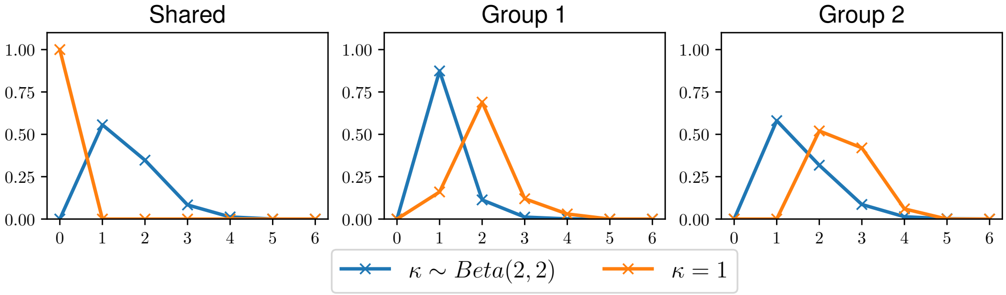


Figure 6.5.1: Posterior distribution of the number of shared unique values and unique values specific to first and second group in Scenario II.

model tend to assume more extreme than those from Camerlenghi et al. (2019). Figure 6.5.1 shows the posterior distribution of the number of shared and private unique values (reconstructed from the cluster allocation variables s_{ij} and the table allocation variables $t_{r\ell}$) in Scenario II, when either $\kappa \sim Beta(2,2)$ or $\kappa = 1$. Also in the he latter case $P(c_1 = c_2 \mid data) = 0$, but the shared component between groups one and two is not recovered, due to the degeneracy issue described in Proposition 6.2.

As the central point of our model is to allow for different random measures to share at least one atom, we test more in detail this scenario. To do so, we simulate 50 different datasets from (6.20), by selecting $\mu_1, \mu_2, \mu_4 \overset{\text{iid}}{\sim} \mathcal{N}(0, 10)$ and $\sigma_1^2, \sigma_2^2, \sigma_4^2 \overset{\text{iid}}{\sim} inv{-}gamma(2,2)$, $w_1 \sim Beta(1,1)$ and setting $\mu_3 = \mu_1, \sigma_3^2 = \sigma_1^2, w_2 = w_1$. In this way we create 50 independent scenarios where the two population share exactly one component and give the same weight to this component. Figure 6.D.1 in the Appendix file reports the scatter plot of the estimated posterior probabilities of $c_1 = c_2$ obtained from the MCMC samples. It is clear that our model recovers the right scenario most of the times. Out of 50 examples, only in four of them $P(c_1 = c_2 \mid data)$ is greater than 0.5, by a visual analysis we see from the plot of the true densities that in those cases the two populations were really similar.

6.5.2 More than two populations

We extend now the simulation study to scenarios with more than two populations. We consider three simulated datasets with four populations each and different clustering structures at the population level. In particular, we use the same scenarios as in Gutiérrez et al. (2019), and simulate $N_i = 100$ points for each population $i = 1, 2, 3, 4$ as follows

- Scenario IV

$$y_{1j}, y_{2k}, y_{3\ell} \overset{\text{iid}}{\sim} \mathcal{N}(0,1) \quad y_{4n} \overset{\text{iid}}{\sim} SN(0,1,1) \quad j,k,\ell,n = 1,\dots,100$$

- Scenario V

$$y_{1j}, y_{4n} \overset{\text{iid}}{\sim} \mathcal{N}(0,1) \quad y_{2k} \overset{\text{iid}}{\sim} \mathcal{N}(0,2.25) \quad y_{3\ell} \overset{\text{iid}}{\sim} \mathcal{N}(0,0.25) \quad j,k,\ell,n = 1,\dots,100$$

- Scenario VI

$$
\begin{aligned}
y_{1j}, y_{2k} &\overset{\text{iid}}{\sim} 0.5\mathcal{N}(0,1) + 0.5\mathcal{N}(5,1) \quad j,k = 1,\dots,100 \\
y_{3\ell} &\overset{\text{iid}}{\sim} 0.5\mathcal{N}(0,1) + 0.5\mathcal{N}(-5,1) \quad \ell = 1,\dots,100 \\
y_{4n} &\overset{\text{iid}}{\sim} 0.5\mathcal{N}(-5,1) + 0.5\mathcal{N}(5,1) \quad n = 1,\dots,100
\end{aligned}
$$

Hence, the true clusters of the label set of the populations, $\{1,2,3,4\}$, are: $\boldsymbol{\rho}_4^{true} = \{\{1,2,3\},\{4\}\}$, $\boldsymbol{\rho}_5^{true} = \{\{1,4\},\{2\},\{3\}\}$ and $\boldsymbol{\rho}_6^{true} = \{\{1,2\},\{3\},\{4\}\}$ for the three scenarios under investigation respectively. By $SN(\xi,\omega,\alpha)$ in Scenario IV we mean the skew-normal distribution with location ξ, scale ω and shape α; in this case, the mean of the distribution is equal to

$$\xi + \omega\frac{\alpha}{1+\alpha^2}\sqrt{\frac{2}{\pi}}.$$

Note that we focus on a different problem than what Gutiérrez et al. (2019) discussed, as they considered testing for multiple treatments against a control. In particular they were concerned about testing the hypothesis of equality in distribution between data coming from different treatments $\boldsymbol{y}_j$ ($j = 2,3,4$ in these scenarios), and data coming from a control group $\boldsymbol{y}_1$. Instead our goal is to cluster these populations based on their distributions.

Observe how the prior chosen for $\boldsymbol{c}$ does not translate directly into a distribution on the partition $\boldsymbol{\rho}$, as it is affected by the so called label switching. Thus, in order to summarize our inference, we post-process our chains and transform the samples $\boldsymbol{c}^{(1)},\dots,\boldsymbol{c}^{(M)}$ from $\boldsymbol{c}$ to samples $\boldsymbol{\rho}^{(1)},\dots,\boldsymbol{\rho}^{(M)}$ from $\boldsymbol{\rho}$. For example we have that $\boldsymbol{c}^{(i)} = (1,1,1,3)$ and $\boldsymbol{c}^{(j)} = (2,2,2,4)$ both get transformed into $\boldsymbol{\rho}^{(i)} = \boldsymbol{\rho}^{(j)} = \{\{1,2,3\},\{4\}\}$.

The posterior probabilities of the true clusters $P(\boldsymbol{\rho}_i = \boldsymbol{\rho}_i^{true} \mid \text{data})$ are estimated using the transformed (as described above) MCMC samples and equal 0.75, 0.95 and 0.99 for the three scenarios respectively. Figure 6.5.2 shows the posterior distribution of $\boldsymbol{\rho}$, and Figure 6.5.3 reports the density estimation of each group, for Scenario IV. Observe how the posterior mode is in $\boldsymbol{\rho}_4^{true}$ but significant mass is given also to the case $\{\{1\},\{2,3\},\{4\}\}$. We believe that this behavior is mainly due to our use of pseudopriors, as it makes the transition between these three states fairly smooth. On the other hand, in Scenario V, where the posterior mass on the true cluster is close to 1, it is clear that such transitions happen very rarely, as the posterior distribution, not shown here, is completely concentrated on $\boldsymbol{\rho}_5^{true}$. Our insight is that the pseudopriors make a transition between two states, say $\boldsymbol{c}^{(j)} = (1,1,3,4)$ and $\boldsymbol{c}^{(j+1)} = (1,2,3,4)$ (or viceversa), more likely when the mixing distributions of population one and two are the same.

We compared the performance of the Metropolised algorithm against the full Gibbs move for the update of $\boldsymbol{c}$, computing the effective sample size (ESS) of the number of population level clusters (i.e. the number of unique values in $\boldsymbol{c}$) over CPU time. We consider two choices for the proposal distribution $p_i(r \mid m)$, namely, the discrete uniform over $\{1,\dots,I\}$ and another discrete alternative, with weights given by

$$p_i(r \mid m) \propto 1 + \left(1 + d^2(\tilde{q}_r, \tilde{q}_m)\right)^{-1} \tag{6.21}$$

where $d^2(\tilde{q}_r,\tilde{q}_m)$ is the squared L^2 distance between the Gaussian mixture represented by $\tilde{q}_r$ and that represented by $\tilde{q}_m$, which are sampled as discussed in Section 6.4. This distance is available in closed form and the formula is reported in Appendix 6.D,

Results for data as in Scenario IV show that the best efficiency is obtained using the full Gibbs update, with an ESS per second of 57.1. The Metropolised sampler with proposal as in (6.21) comes second, yielding an ESS per second of 34.1 while the Metropolised sampler with uniform proposal is the worst performer with an ESS per second of 12.8. Hence, even when the number of groups is not enormous, the good performance of the Metropolised

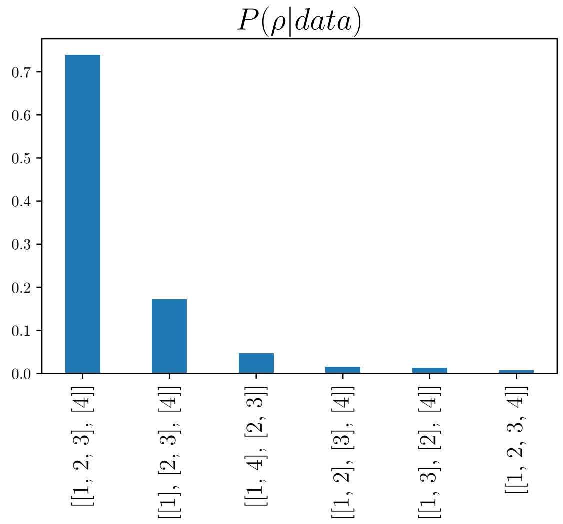


Figure 6.5.2: Posterior probability of $\boldsymbol{\rho}$ for Scenario IV.

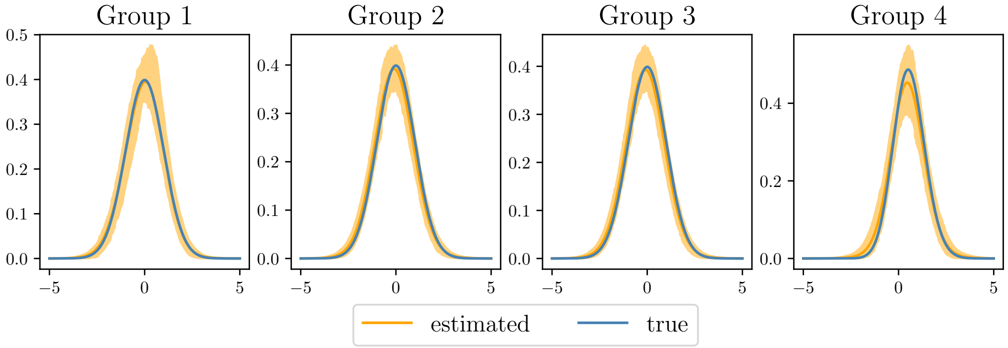


Figure 6.5.3: Density estimates and pointwise 95% posterior credible intervals for Scenario IV.

sampler is clear. Preliminary analysis showed how the Metropolised sampler outperforms the full Gibbs one as the number of groups increases.

Finally, we test how our algorithm performs when the number of populations increases significantly. We do so by generating 100 populations in Scenario VII as follows:

$$\begin{aligned}
y_{ij} &\overset{\text{iid}}{\sim} 0.5\mathcal{N}(-5,1) + 0.5\mathcal{N}(5,1) \quad & i = 1,\dots,20 \\
y_{ij} &\overset{\text{iid}}{\sim} 0.5\mathcal{N}(-5,1) + 0.5\mathcal{N}(0,1) \quad & i = 21,\dots,40 \\
y_{ij} &\overset{\text{iid}}{\sim} 0.5\mathcal{N}(0,1) + 0.5\mathcal{N}(5,0.1) \quad & i = 41,\dots,60 \\
y_{ij} &\overset{\text{iid}}{\sim} 0.5\mathcal{N}(-10,1) + 0.5\mathcal{N}(0,1) \quad & i = 61,\dots,80 \\
y_{ij} &\overset{\text{iid}}{\sim} 0.1\mathcal{N}(-10,1) + 0.9\mathcal{N}(0,1) \quad & i = 81,\dots,100.
\end{aligned}$$

Thus, full exchangeability holds within populations $\{1,\dots,20\}$, $\{21,\dots,40\}$, $\{41,\dots,60\}$, $\{61,\dots,80\}$ and $\{81,\dots,100\}$ but not between these five groups. For each population i, 100 datapoints were sampled independently.

To compute posterior inference, we run the Metropolised sampler with proposal (6.21). To get a rough idea of the computational costs associated to this large simulated dataset, we report that running the full Gibbs sampler would have required more than 24 hours on a

32-core machine (having parallelized all the computations which can be safely parallelized), while the Metropolised sampler ran in less than 3 hours on a 6-core laptop.

As a summary of the posterior distribution of the random partition $\boldsymbol{\rho}_{100}$, we compute the posterior similarity matrix $[P(c_i = c_j \mid data)]_{i,j=1}^{I}$. Estimates of these probabilities are straightforward to obtain using the output of the MCMC algorithm. Figure 6.5.4 shows the posterior similarity matrix as well as the density estimates of five different populations. It is clear that the clustering structure of the populations is recovered perfectly and that the density estimates are coherent with the true ones.

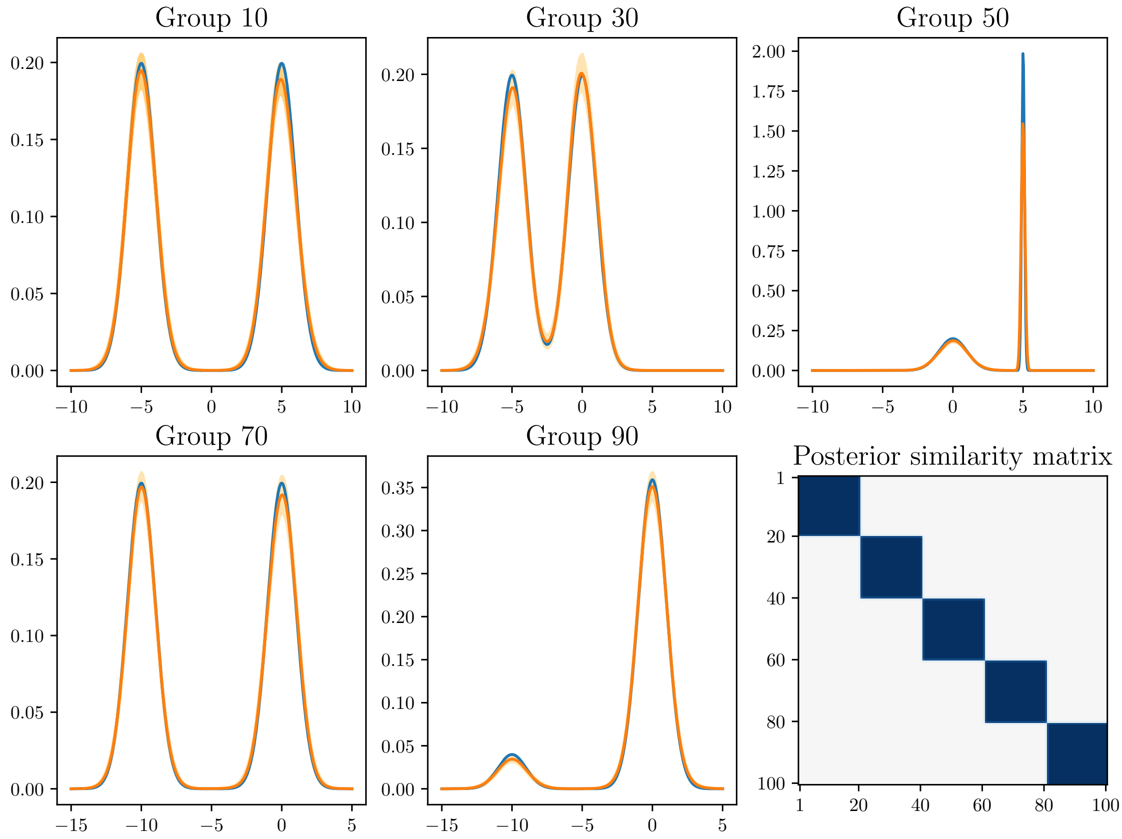


Figure 6.5.4: Density estimates (orange line), pointwise 95% posterior credible intervals (orange bands), true data generating densities (blue line) for groups 10, 30, 50, 70 and 90 and posterior similarity matrix (bottom right, white corresponds to 0.0 and dark blue to 1.0) in Scenario VII.

6.6 Chilean grades dataset

The School of Mathematics at Pontificia Universidad Católica de Chile teaches many undergraduate courses to students from virtually all fields. When the number of students exceeds a certain maximum pre-established quota, several sections are formed, and courses are taught in parallel. There is a high degree of preparation in such cases, so as to guarantee that courses cover the same material and are coordinated to function as virtual copies of each other. In such cases, only the instructor changes across sections, but all materials related to the courses are the same, including exams, homework, assignments, projects, etc., and there is a shared team of graders that are common to all the parallel sections. According to the rules, every student gets a final grade on a scale from 1.0 to 7.0, using

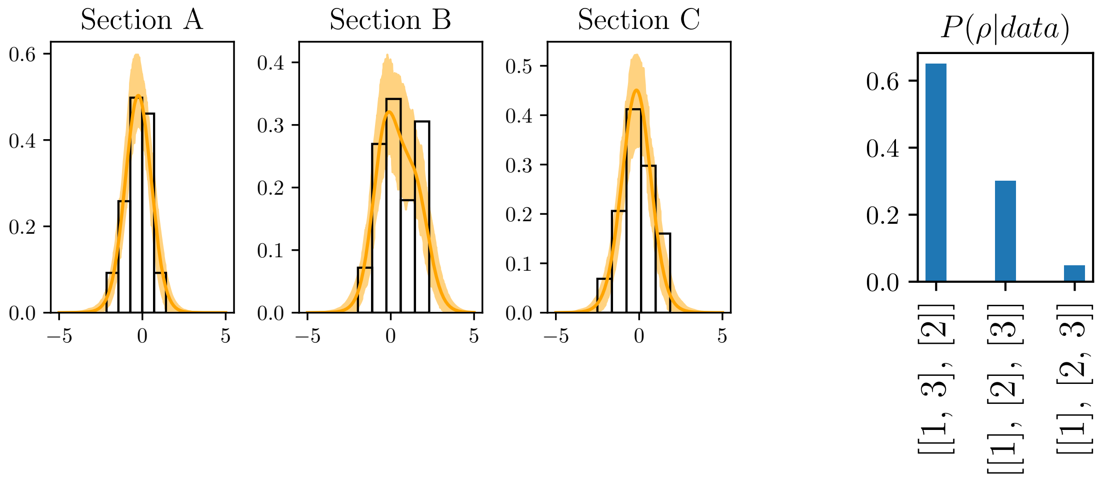


Figure 6.6.1: Density estimates and pointwise 95% posterior credible intervals for the three groups (left); posterior distribution of the clusters (right).

one decimal place, where 4.0 is the minimum passing grade. We consider here the specific case of a version of Calculus II, taught in parallel to three different sections (A, B and C) in a recent semester. Our main goal here is to examine the instructor effectiveness, by comparing the distributions of the final grades obtained by each of the three populations (sections). The sample sizes of these populations are 76, 65 and 50 respectively.

A possible way to model these data could be to employ a truncated normal distribution as the kernel in (6.2). However since our primary interest is to investigate the homogeneity of the underlying distributions and not to perform density estimates, we decided to first add a small amount of zero-mean Gaussian noise, with variance 0.1 to the data (i.e. 'jittering') and then proceeded to standardize the whole dataset, by letting $y_{ij}^{new} = (y_{ij} - \bar{y})/s_y$, where $\bar{y} = (\sum_{ij} y_{ij})/(\sum_i N_i)$ and $s_y^2 = (\sum_{ij}(y_{ij} - \bar{y})^2)/(\sum_i N_i - 1)$ are the global sample mean and variance, respectively. In the sequel, index $i = 1, 2, 3$ denotes sections A, B and C, respectively, as described above.

Figure 6.6.1 reports density estimates in all groups (i.e. posterior density means evaluated on a fixed grid of points and pointwise 95% posterior credible intervals at each point x in the grid), as well as the posterior distribution of the random partition $\boldsymbol{\rho}$, obtained from the posterior distribution of $\boldsymbol{c}$, getting rid of the label switching in a post-processing step (see also Section 6.5.2). From Figure 6.6.1 we see that the posterior distribution of $\boldsymbol{\rho}$ gives high probability to the case of the three groups being all different as well as to the case when the first and third groups are homogeneous but different from the second one. This is in accordance with a visual analysis of the observed and estimated densities.

We considered several functionals of the random population distribution $\tilde{q}_{c_i}$ (see (6.3)) for $i = 1, 2, 3$. Recall that, according to notation in (6.1), $\tilde{q}_{c_i} = \tilde{p}_i$. First of all, we consider the mean and variance functionals of the random density $p_i(y) = \int_\Theta k(y|\theta)\tilde{q}_{c_i}(d\theta) = \int_\Theta k(y|\theta)\tilde{p}_i(d\theta)$, for each $i = 1, 2, 3$. Observe how they are functionals of the random probability $\tilde{q}_{c_i} = \tilde{p}_i$. Moreover, since Figure 6.6.1 seems to suggest that the three groups differ mainly due to their different asymmetries, we considered two more functionals of $\tilde{p}_i$, i.e. two indicators of skewness: Pearson's moment coefficient of skewness sk and the measure of skewness with respect to the mode γ_M proposed by Arnold and Groeneveld (1995). Pearson's moment coefficient of skewness of the random variable T is defined as $\text{sk} = \mathbb{E}[((T - \mathbb{E}(T))/\sqrt{\text{Var}(T)})^3]$, while the measure of skewness with respect to the mode as $\gamma_M = 1 - 2F_T(M_T)$, where M_T is the mode of T and F_T denotes its distribution function. The last functional of $\tilde{p}_i$ we consider is the probability, under the density

Section	μ_i	σ_i^2	sk$_i$	γ_{Mi}	P_{4i}
A	-0.264	0.671	120.84	-0.01	0.53
B	0.438	1.428	-64.86	0.292	0.71
C	-0.171	0.943	55.60	-0.01	0.56

Table 6.6.1: Posterior means of functionals $\mu_i, \dots, P_{4i}$ of the population density p_i for each Section A ($i = 1$), B ($i = 2$) and C ($i = 3$) in the Chilean grades dataset. All the functionals refer to standardized data $\{y_{ij}^{new} = (y_{ij} - \bar{y})/s_y\}$.

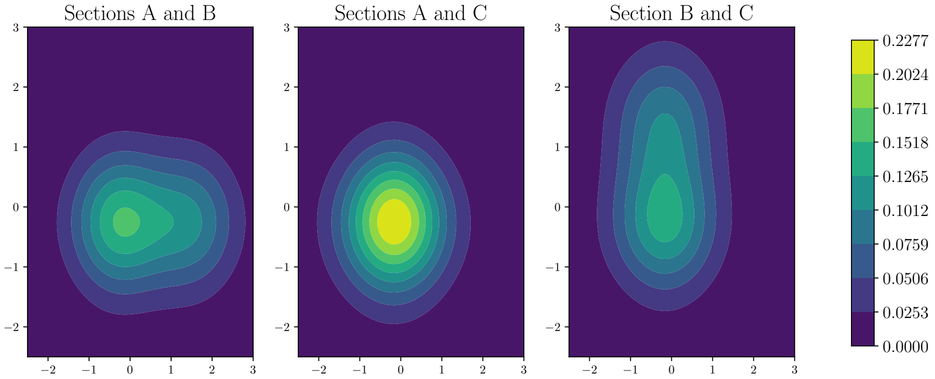


Figure 6.6.2: Posterior means of (p_i, p_ℓ), $i \neq \ell$, $i, \ell = A, B, C$, evaluated on a fixed grid in $\mathbb{R}^2$ for the Chilean grades dataset.

$p_i(y) = \int_\Theta k(y|\theta)\tilde{p}_i(d\theta)$ of getting a passing grade (≥ 4.0 before normalization), that is

$$P_{4i} = \int_{(4-\bar{y})/s_y}^{+\infty} p_i(y)dy.$$

Table 6.6.1 shows the posterior mean of the functionals μ_i, σ_i^2 (mean and variance functionals), sk_i, γ_{Mi} and P_{4i} of p_i, for $i = 1, 2, 3$. To be clear, the posterior mean of the mean functional μ_1 is computed as

$$\frac{1}{M}\sum_{\ell=1}^{M} \mu_1^{(\ell)} = \frac{1}{M}\sum_{\ell=1}^{M} \mathbb{E}[y \,|\, \tilde{p}_1^{(\ell)}] = \frac{1}{M}\sum_{\ell=1}^{M} \left(\int_{\mathbb{R}} y p_1^{(\ell)}(y)dy \right),$$

where M is the MCMC sample size, and the superscript (ℓ) attached to a random variable denotes its value at the ℓ–th MCMC iteration.

In agreement with the posterior distribution of the partition $\boldsymbol{\rho}$, for all the functionals considered we observed close values for sections A and C, while both differ significantly from the values for section B. In summary, we conclude that section B presents a heavier right tail than sections A and C, hence it is characterized by a higher mean (positive) and also more spread across the range. Section B shows a larger (estimated) value for P_4, i.e. students in section B are more likely to pass the exam than their colleagues from the other sections. This seems to suggest that a higher concentration of good students (with high grades) was present in Section B, compared to A and C, possibly combined with a higher effectiveness of the instructor in this Section.

We also computed the pairwise L^1 distances between the estimated densities in the populations. If $\bar{p}_i$ denotes the estimated density (posterior mean of p_i evaluated in a grid of points) for each population, we found $d(\bar{p}_A, \bar{p}_B) = 0.56$, $d(\bar{p}_A, \bar{p}_C) = 0.15$ and

$d(\bar{p}_B, \tilde{p}_C) = 0.44$. This confirms once again that the estimated densities for section A and C are closer than when comparing sections A and B and sections B and C.

To end the analysis, we show in Figure 6.6.2 estimated couples of densities (p_i, p_ℓ), $i \neq \ell$, $i, \ell = 1, 2, 3$, i.e. the posterior mean of (p_i, p_ℓ), evaluated on a fixed grid in $\mathbb{R}^2$. While sections A and C look independent (central panel in Figure 6.6.2), the (posterior) propensity of section B to get higher grades is confirmed in the left and right panels in Figure 6.6.2.

6.7 Discussion

Motivated by the traditional problem of testing homogeneity across I different groups or populations, we have presented a model that is able to not only address the problem but also to perform a cluster analysis of the groups. The model is built on a prior for the population distributions that we termed the semi-hierarchical Dirichlet process, and it was shown to have good properties and also to perform well in synthetic and real data examples, also in case of $I = 100$ groups. One of the driving features of our proposal was to solve the degeneracy limitation of nested constructions that has been pointed out by Camerlenghi et al. (2019). The crucial aspect of the semi-HDP that solves this problem was described using the metaphor of a food court of Chinese restaurants with common and private dining area. The hierarchical construction introduces a random partition at the population level, which allows for identifying possible clusters of internally homogeneous groups.

Our examples focus on unidimensional data, though extensions to multivariate responses can be straightforwardly accommodated in our framework. However, scaling with respect to data dimension is not a property we claim to have. In fact, this is a situation shared with any type of hierarchical mixture models.

We studied support properties of the semi-HDP and also the posterior asymptotic behavior of the Bayes factor for the homogeneity test when $I = 2$, as posed within the proposed hierarchical construction. We showed that the Bayes factor has the appropriate asymptotic behavior under the alternative hypothesis of partial exchangeability, but a final answer under the assumption of truly exchangeable data is still pending. The lack of asymptotic guarantees is not at all specific to our case. In fact, this situation is rather common to all model selection problems when the hypothesis are not well separated and at least one of the two models under comparison is 'truly' nonparametric, as, for instance, in Bhattacharya and Dunson (2012) and Tokdar and Martin (2019). Indeed, as discussed in Tokdar and Martin (2019), it is not even clear if in such cases the need for an upper bound on the prior mass under the more complex model is a natural requirement or rather a technical one. More generally, intuition about BFs (at least in parametric cases) is that they tend to favor the more parsimonious model. In the particular context described in Section 6.3.3, model M_1 can be regarded as a degenerate case of model M_2, even though they are 'equally complicated'. In this case, the above intuition evaporates, since technically, embedding one model in the other is still one infinite-dimensional model contained in another infinite-dimensional model, and it is probably meaningless to ask which model is 'simpler'. Under this scenario exploratory use of discrepancy measures, such as those discussed in Gelman et al. (1996), may offer some guidance.

In the simulation studies presented, our model always recovers the true latent clustering among groups, thus providing empirical evidence in favor of our model to perform homogeneity tests. We provide some practical suggestions when the actual interest is on making this decision. Our insight is that in order to prove asymptotic consistency of the Bayes factor, one should introduce explicit separation between the competing hypotheses. One possible way to accomplish this goal is, for example, by introducing some kind of repulsion among the mixing measures $\tilde{q}_i$'s in the model. This point will be focus of further study.

Appendix

6.A Proofs

Proof of **Proposition 6.1.**

Consider $I = 2$ for ease of exposition. We aim at showing that under suitable choices of G_0 and G_{00}, the vector of random probability measures $(\tilde{p}_1, \tilde{p}_2)$, where $\tilde{p}_i = \tilde{q}_{c_i}$ has full support on $N_\Theta \times N_\Theta$.

This means that for every couple of distributions $(g_1, g_2) \in N_\Theta \times N_\Theta$, every weak neighborhood $W_1 \times W_2$ of (g_1, g_2) receives non null probability. In short, this condition entails $\pi_{\tilde{\boldsymbol{p}}}(W_1 \times W_2) > 0$. Since $\tilde{p}_i = \tilde{q}_{c_i}$, we have that

$$\pi_{\tilde{\boldsymbol{p}}}(W_1 \times W_2) = \sum_{l,m=1}^{2} \pi_{\tilde{q}_l, \tilde{q}_m}(W_1 \times W_2) \pi_c(l, m) > \pi_{\tilde{q}_1, \tilde{q}_2}(W_1 \times W_2) \pi_c(1, 2).$$

Hence, since we are assuming that $\pi_c(l, m) > 0$ for all l, m, it is sufficient to show that $\pi_{\tilde{q}_1, \tilde{q}_2}$, that is the measure associated to the SemiHDP prior with $I = 2$, has full weak support.

In the following, with a slight abuse of notation we denote by $\pi_{\tilde{q}_1, \tilde{q}_2 \,|\, \tilde{q}_0}(W_1 \times W_2)$ the measure associated to the SemiHDP prior, conditional to a particular value of $\tilde{q}_0$. We distinguish three cases: $\kappa = 1$, $0 < \kappa < 1$ and $\kappa = 0$. The case $\kappa = 1$ is trivial, since $\tilde{q}_1$ and $\tilde{q}_2$ are marginally independently distributed with Dirichlet process prior, so that $\pi_{\tilde{q}_1, \tilde{q}_2}(W_1 \times W_2) = \mathcal{D}_{\alpha G_0}(W_1) \mathcal{D}_{\alpha G_0}(W_2) > 0$ as long as G_0 has full support in Θ (see, for example, Ghosal and Van der Vaart, 2017).

Secondly consider $0 < \kappa < 1$, we show that as long as G_0 has full support, then also $\pi_{\boldsymbol{G}}$ will have full support, regardless of the properties of G_{00}. We have

$$\pi_{\tilde{q}_1, \tilde{q}_2}(W_1 \times W_2) = \int_{N_\Theta} \pi_{\tilde{q}_1, \tilde{q}_2 \,|\, \tilde{q}_0}(W_1 \times W_2) \mathcal{L}(d\tilde{q}_0) = \int_{N_\Theta} \mathcal{D}_{\alpha \tilde{p}}(W_1) \mathcal{D}_{\alpha \tilde{p}}(W_2) \mathcal{L}(d\tilde{q}_0). \tag{6.22}$$

Now observe that if G_0 has full support, also $\tilde{p} = \kappa G_0 + (1 - \kappa)\tilde{q}_0$ will have full support, for any value of $\tilde{q}_0$. Hence by the properties of the Dirichlet Process, we get that $\pi_{\tilde{q}_1, \tilde{q}_2}(W_1 \times W_2) > 0$ since the integrand in (6.22) is bounded away from zero.

The case $\kappa = 0$ is more delicate and requires additional work. We follow the path outlined in De Blasi et al. (2013), extending it to our hierarchical case. In the following, let $\mathbb{S}^m$ denote the $m - 1$ dimensional simplex, i.e.

$$\mathbb{S}^m := \{(z_1, \dots, z_m) \in \mathbb{R}^m \ : \ 0 \le z_h \le 1, \ h = 1, \dots, m, \ \sum_{h=1}^{m} z_h = 1\}$$

Let d_w denote the Prokhorov metric on N_Θ, which, as it is well known, metrizes the topology of the weak convergence on N_Θ. Moreover, being Θ separable, (N_Θ, d_w) is separable as well and the set of discrete measures with a finite number of point masses is dense in N_Θ.

Hence, for any (g_1, g_2) and any $\epsilon > 0$, there exist two discrete measures with weights $\boldsymbol{p}^{(i)} \in \mathbb{S}^{k_i}$ and points $\boldsymbol{x}^{(i)} \in \Theta^{k_i}$ for $i = 1, 2$ such that $d_w(\tilde{q}_{\boldsymbol{p}^{(i)},\boldsymbol{x}^{(i)}}, g_i) < \epsilon$, where $\tilde{q}_{\boldsymbol{p}^{(i)},\boldsymbol{x}^{(i)}} = \sum_k p_k^{(i)} \delta_{x_k^{(i)}}$. The difficulty when $\kappa = 0$ is that conditionally on $\tilde{q}_0$, the measure $\mathcal{D}_{\alpha\tilde{q}_0}$ does not have full weak support. Indeed, its support is concentrated on the measures that have the same atoms of $\tilde{q}_0$. The proof will proceed as follows: start by defining weak neighborhoods W_i of $\tilde{q}_{\boldsymbol{p}^{(i)},\boldsymbol{x}^{(i)}}$ by looking at neighborhoods of their weights $\boldsymbol{p}^{(i)}$ (U_i) and atoms $\boldsymbol{x}^{(i)}$ (V_i). Secondly, we join these neighborhoods. If $\tilde{q}_0(\omega)$ belongs to this union (and this occurs with positive probability), we guarantee that the atoms of both of $\tilde{q}_1$ and $\tilde{q}_2$, that are shared with $\tilde{q}_0$, are suited to approximate both $\tilde{q}_{\boldsymbol{p}^{(i)},\boldsymbol{x}^{(i)}}$, $i = 1, 2$. Hence, by the properties of the Dirichlet Process one gets the support property.

More in detail, define the sets

$$V_i(\delta) = \{\boldsymbol{x}_i \in \Theta^{k_i} \ s.t. \ |x_{ij} - x_j^{(i)}| < \delta\}, i = 1, 2$$

and let $V = V_1 \cup V_2$. Then we operate a change of index by concatenating $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$, and call it $\boldsymbol{x}^*$, i.e. $\boldsymbol{x}^* = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}]$. Hence we characterize the set V as

$$V(\delta) = \{\boldsymbol{x} \in \Theta^{k_1+k_2} \ s.t \ |x_j - x_j^*| < \delta\}.$$

Secondly, define $\boldsymbol{p}^*$ by concatenating $\boldsymbol{p}^{(1)}$ and and $\boldsymbol{p}^{(2)}$: $\boldsymbol{p}^* = [\boldsymbol{p}^{(1)}, \boldsymbol{p}^{(2)}]$ and let

$$U_1(\eta) = \{\boldsymbol{p} \in \mathbb{S}^{k_1+k_2} \ s.t. \ |p_j - p_j^*| < \eta \text{ for } j = 1, \ldots, k_1, \ |p_j - 0| < \eta \text{ elsewhere}\}$$
$$U_2(\eta) = \{\boldsymbol{p} \in \mathbb{S}^{k_1+k_2} \ s.t. \ |p_j - p_j^*| < \eta \text{ for } j = k_1 + 1, \ldots, k_1 + k_2, \ |p_j - 0| < \eta \text{ elsewhere}\}$$

Finally, define the following neighborhoods

$$W_i := \{ \sum_{j=1}^{k_1+k_2} p_j \delta_{x_j} \text{ for any } \boldsymbol{p} \in U_i, \text{ and any } \boldsymbol{x} \in V\}, i = 1, 2$$
$$W_0 := \{ \sum_{j=1}^{k_1+k_2} p_j \delta_{x_j} \text{ for any } \boldsymbol{p} \in \mathbb{S}^{k_1+k_2}, \text{ and any } \boldsymbol{x} \in V\}.$$

This means that the V_i sets are the neighborhoods of the atoms $\boldsymbol{x}_i$ that are well suited to approximate $\tilde{q}_{\boldsymbol{p}^{(i)},\boldsymbol{x}^{(i)}}$ and V is their union. The sets U_i, $i = 1, 2$, instead, are related to the weights of $\tilde{q}_{\boldsymbol{p}^{(i)},\boldsymbol{x}^{(i)}}$. In particular, each U_i is constructed in such a way to approximate well $\boldsymbol{p}^{(i)}$ (a vector in $\mathbb{S}^{k_i}$) with a vector of weights in $\mathbb{S}^{k_1+k_2}$. This is necessary because if $\tilde{q}_0$ has support points in V, so will do the draws $\tilde{q}_1$ and $\tilde{q}_2$ from $\mathcal{D}_{\alpha\tilde{q}_0}$. However, by assigning a negligible weight in U_1 to the atoms $\boldsymbol{x}^{(2)}$ and vice-versa for the atoms $\boldsymbol{x}^{(1)}$ in U_2, we guarantee that the probability measures in W_i constitute a weak neighborhood of $\tilde{q}_{\boldsymbol{p}^{(i)},\boldsymbol{x}^{(i)}}$ for each $i = 1, 2$.

From De Blasi et al. (2013), it is sufficient to show that $\pi_{\tilde{q}_1,\tilde{q}_2}(W_1 \times W_2) > 0$ since for appropriate choices of η and δ one has that $d_w(F_1, g_1) + d_w(F_2, g_2) < \epsilon$ for all choices of $F_1 \in W_1$ and $F_2 \in W_2$. Hence

$$\begin{aligned}\pi_{\tilde{q}_1,\tilde{q}_2}(W_1 \times W_2) = \int_{N_\Theta} \pi_{\tilde{q}_1,\tilde{q}_2 \,|\, \tilde{q}_0}(W_1 \times W_2)\mathcal{L}(d\tilde{q}_0) &\geq \int_{W_0} \pi_{\tilde{q}_1,\tilde{q}_2 \,|\, \tilde{q}_0}(W_1, W_2)\mathcal{L}(d\tilde{q}_0) \\ &= \int_{W_0} \mathcal{D}_{\alpha\tilde{q}_0}(W_1)\mathcal{D}_{\alpha\tilde{q}_0}(W_2)\mathcal{L}(d\tilde{q}_0)\end{aligned}$$

Now observe that for any $\tilde{q}_0 \in W_0$, we have that $\mathcal{D}_{\alpha\tilde{q}_0}(W_i) > 0$. This follows again from the properties of the Dirichlet process, since for any value of $\tilde{q}_0(\omega)$, there exists a non-empty set $\widetilde{W}_i \subset W_i$, $\widetilde{W}_i = \{\widetilde{F}_i \in W_i : \ supp(\widetilde{F}_i) \subset supp(\tilde{q}_0)\}$. Hence $\pi_{F_1,F_2 \,|\, \tilde{q}_0}(W_1 \times W_2) \geq$

$\pi_{F_1,F_2 \,|\, \tilde{q}_0}(\widetilde{W}_1 \times \widetilde{W}_2) > 0$, since the Dirichlet process gives positive probability to the weak neighborhoods of measures whose support is contained in the support of its base measure, i.e. $\tilde{q}_0$. □

Proof of **Covariance of the semi-HDP**.
If $(\tilde{q}_1, \tilde{q}_2) \sim semiHDP(\alpha, \gamma, \kappa, G_0, G_0)$, then

$$\begin{aligned}
\operatorname{cov}(\tilde{q}_1(A)\tilde{q}_2(B)) &= \mathbb{E}\left[\tilde{q}_1(A)\tilde{q}_2(B)\right] - \mathbb{E}\left[\tilde{q}_1(A)\right]\mathbb{E}\left[\tilde{q}_2(B)\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\tilde{q}_1(A)\tilde{q}_2(B) \mid \tilde{q}\right]\right] - \mathbb{E}\left[\mathbb{E}\left[\tilde{q}_1(A) \mid \tilde{q}_0\right]\right]\mathbb{E}\left[\mathbb{E}\left[\tilde{q}_2(B) \mid \tilde{q}_0\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\tilde{q}_1(A) \mid \tilde{q}_0\right]\mathbb{E}\left[\tilde{q}_2(B) \mid \tilde{q}_0\right]\right] - G_0(A)G_0(B) \\
&= \mathbb{E}\left[\tilde{p}(A)\tilde{p}(B)\right] - G_0(A)G_0(B) \\
&= \kappa^2 G_0(A)G_0(B) + \kappa(1-\kappa)G_0(A)\mathbb{E}\left[\tilde{q}_0(B)\right] + \kappa(1-\kappa)G_0(B)\mathbb{E}\left[\tilde{q}_0(A)\right] \\
&\quad + (1-\kappa)^2\mathbb{E}\left[\tilde{q}_0(A)\tilde{q}_0(B)\right] - G_0(A)G_0(B) \\
&= (1-\kappa)^2\mathbb{E}\left[\tilde{q}_0(A)\tilde{q}_0(B)\right] - (1-\kappa)^2 G_0(A)G_0(B) \\
&= (1-\kappa)^2\operatorname{cov}(\tilde{q}_0(A), \tilde{q}_0(B)) = \frac{(1-\kappa)^2}{1+\gamma}\left(G_0(A \cap B) - G_0(A)G_0(B)\right).
\end{aligned}$$

The last equality follows because $\tilde{q}_0$ is a Dirichlet process. □

Higher order moments.
To compute higher order moments, we make use of a result from Argiento et al. (2019). Let $\tilde{q}_1 \,|\, \tilde{p} \sim \mathcal{D}_{\alpha\tilde{p}}$ as in (6.5) - (6.7); then one has, for any set $A \in \mathcal{B}(\Theta)$:

$$\mathbb{E}[\tilde{q}_1(A)^n \,|\, \tilde{q}_0] = \sum_{t=1}^{n} \tilde{p}(A)^t P(K_n = t),$$

where K_n is the random variable representing the number of *clusters* in a sample of size n; see (15) in Argiento et al. (2019). If, as in our case, the base measure is not absolutely continuous, the term clusters might be misleading as they do not coincide with the unique values in the sample, but rather with the number of the tables in the Chinese restaurant process. In the following we refer to cluster or table interchangeably. Hence, we have:

$$\begin{aligned}
\mathbb{E}[\tilde{q}_1(A)^n] = \mathbb{E}[\mathbb{E}[\tilde{q}_1(A)^n \,|\, \tilde{q}_0]] &= \mathbb{E}\left[\sum_{t=1}^{n} \tilde{p}(A)^t P(K_n = t)\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{n} P(K_n = t) \sum_{h=0}^{t} \binom{t}{h} (\kappa G_0(A))^{t-h} \times ((1-\kappa)\tilde{q}_0(A))^h\right] \\
&= \sum_{t=1}^{n} P(K_n = t) \sum_{h=0}^{t} \binom{t}{h} (\kappa G_0(A))^{t-h} (1-\kappa)^h \mathbb{E}[\tilde{q}_0(A)^h] \\
&= \sum_{t=1}^{n} P(K_n = t) \sum_{h=0}^{t} \binom{t}{h} (\kappa G_0(A))^{t-h} (1-\kappa)^h \sum_{m=1}^{h} G_{00}(A) P(\widetilde{K}_h = m),
\end{aligned}$$

where $\widetilde{K}_h$ is the number of clusters from a sample of size h from the DP $\tilde{q}_0$. Moreover, if we assume $G_0 = G_{00}$ we get

$$\mathbb{E}[\tilde{q}_1(A)^n] = \sum_{t=1}^{n} P(K_n = t) \sum_{h=0}^{t} \binom{t}{h} \kappa^{t-h} (1-\kappa)^h \sum_{m=1}^{h} G_0(A)^{t-h+m} P(K_h = m).$$

Figure 6.A.1 shows the effect of the parameter κ over $\mathbb{E}[\tilde{q}_1(A)^3]$ for various values of $G_0(A)$. The limiting cases of the standard Dirichlet process and the Hierarchical Dirichlet Process are recovered when $\kappa = 1$ and $\kappa = 0$ respectively.

Proof of **Proposition 6.2**.
Indicating with τ_j the shared unique values between $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, and with θ^*_{ij} the unique values in sample $\boldsymbol{\theta}_i$ that are specific to group i, i.e. not shared, the pEPPF, given $\boldsymbol{c}$, can be written as:

$$\Pi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1, \boldsymbol{q}_2 | \boldsymbol{c}) = \int_{\Theta^k} \mathbb{E}\left[\prod_{j=1}^{k_1} \tilde{q}_{c_1}^{n_{1j}}(d\theta^*_{1j}) \prod_{j=1}^{k_2} \tilde{q}_{c_2}^{n_{2j}}(d\theta^*_{2j}) \prod_{j=1}^{k_0} \tilde{q}_{c_1}^{q_{1j}}(d\tau_j) \tilde{q}_{c_2}^{q_{2j}}(d\tau_j)\right].$$

See (23) in Camerlenghi et al. (2019). Marginalizing out $\boldsymbol{c}$ we obtain that:

$$\Pi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1, \boldsymbol{q}_2) = \sum_{l,m=1}^{2} \pi_{\boldsymbol{c}}(\boldsymbol{c} = (l,m)) \Pi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1, \boldsymbol{q}_2 | \boldsymbol{c} = (l,m)).$$

The cases $\boldsymbol{c} = (1,1)$ and $\boldsymbol{c} = (2,2)$ can be easily managed as it corresponds to full exchangeability and the EPPF corresponding to those cases is already available. Hence,

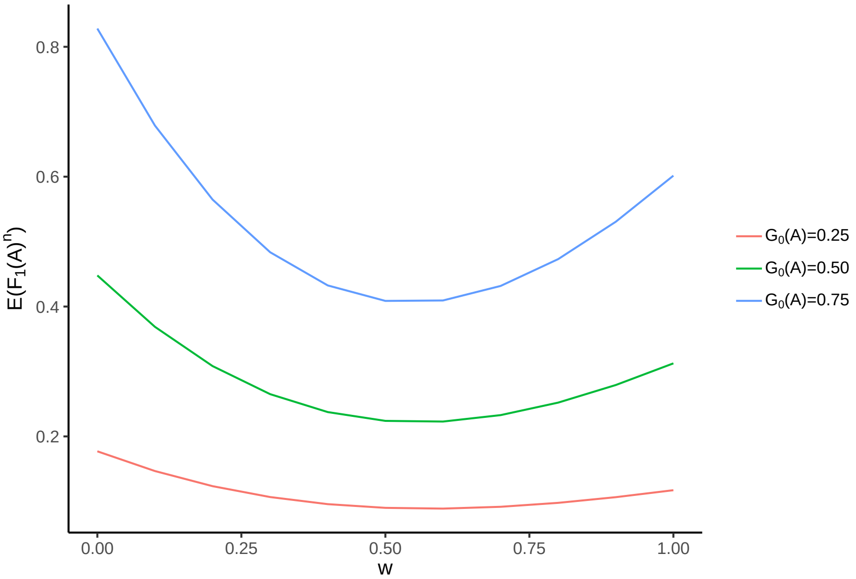


Figure 6.A.1: 3-rd moment of $\tilde{q}_1(A)$ for increasing values of κ and various values of $G_0(A)$.

let us consider the case when $\boldsymbol{c} = (1, 2)$, as the case $\boldsymbol{c} = (2, 1)$ will be identical because the $\tilde{q}_i$'s are iid.

$$\Pi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1, \boldsymbol{q}_2 | \boldsymbol{c} = (1,2)) = \int_{\Theta^k} \mathbb{E}\left[\prod_{j=1}^{k_1} \tilde{q}_1^{n_{1j}}(d\theta^*_{1j}) \prod_{j=1}^{k_2} \tilde{q}_2^{n_{2j}}(d\theta^*_{2j}) \prod_{j=1}^{k_0} \tilde{q}_1^{q_{1j}}(d\tau_j)\tilde{q}_2^{q_{2j}}(d\tau_j)\right]$$
$$= \int_{\Theta^k} \mathbb{E}\left[\prod_{j=1}^{k_1} \tilde{q}_1^{n_{1j}}(d\theta^*_{1j}) \prod_{j=1}^{k_0} \tilde{q}_1^{q_{1j}}(d\tau_j)\right] \mathbb{E}\left[\prod_{j=1}^{k_2} \tilde{q}_2^{n_{2j}}(d\theta^*_{2j}) \prod_{j=1}^{k_0}(d\tau_j)\tilde{q}_2^{q_{2j}}(d\tau_j)\right]$$

since $\tilde{q}_1$ and $\tilde{q}_2$ are independent. The first expected value is the joint probability of $\Pi_{k_1+k_0}^{N_1}$ (the EPPF of a partition of N_1 objects into $k_1 + k_0$ groups with vectors of frequencies $\boldsymbol{n}_1, \boldsymbol{q}_1$) and the set of unique values is denoted by $(x_{11}, \ldots, x_{1k_1}, \tau_1, \ldots \tau_{k_0})$. Similarly for the second expected value. Because $\tilde{q}_1 \sim \mathcal{D}_{\alpha G_0}$, we can rewrite the expected value as:

$$\mathbb{E}\left[\prod_{j=1}^{k_1} \tilde{q}_1^{n_{1j}}(d\theta^*_{1j}) \prod_{j=1}^{k_0} \tilde{q}_1^{q_{1j}(d\tau_j)}\right]$$
$$= \frac{\alpha_1^{k_1+k_0}\Gamma(\alpha_1)}{\Gamma(\alpha_1 + N_1)} \prod_{j=1}^{k_1} \Gamma(n_{1j}) \prod_{j=1}^{k_0} \Gamma(q_{1j}) \prod_{j=1}^{k_1} G_0(d\theta^*_{1j}) \prod_{j=1}^{k_0} G_0(d\tau_j).$$

Hence, we have that

$$
\begin{aligned}
&\Pi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1, \boldsymbol{q}_2 | \boldsymbol{c} = (1,2)) = \\
&\quad = \int_{\Theta^k} \mathbb{E}\left[\prod_{j=1}^{k_1} \tilde{q}_1^{n_{1j}}(d\theta^*_{1j}) \prod_{j=1}^{k_0} \tilde{q}_1^{q_{1j}}(d\tau_j)\right] \mathbb{E}\left[\prod_{j=1}^{k_2} \tilde{q}_2^{n_{2j}}(d\theta^*_{2j}) \prod_{j=1}^{k_0}(d\tau_j)\tilde{q}_2^{q_{2j}}(d\tau_j)\right] \\
&\quad = \frac{\alpha_1^{k_1+k_0}\Gamma(\alpha_1)}{\Gamma(\alpha_1+N_1)} \frac{\alpha_2^{k_2+k_0}\Gamma(\alpha_2)}{\Gamma(\alpha_2+N_2)} \prod_{j=1}^{k_1}\Gamma(n_{1j}) \prod_{j=1}^{k_2}\Gamma(n_{2j}) \prod_{j=1}^{k_0}\Gamma(q_{1j})\Gamma(q_{2j}) \\
&\qquad\qquad \times \int_{\Theta^k} \prod_{j=1}^{k_1} G_0(d\theta^*_{1j}) \prod_{j=1}^{k_2} G_0(d\theta^*_{2j}) \prod_{j=1}^{k_0} G_0(d\tau_j) G_0(d\tau_j).
\end{aligned}
$$

Looking at the last integral, we can see that this is clearly 0 unless $k_0 = 0$, in fact, consider $k_0 = 1$:

$$
\int_{\Theta^{k-1}} \prod_{j=1}^{k_1} G_0(d\theta^*_{1j}) \prod_{j=1}^{k_2} G_0(d\theta^*_{2j}) \int_{\Theta} G_0(dz) G_0(dz)
$$

and observe that the last integral is integrating the product measure $G_0 \times G_0$ on the straight line $y = x$, resulting thus in 0.

Summing up, if $k_0 = 0$ we get:

$$
\begin{aligned}
\Pi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1, \boldsymbol{q}_2) =& \pi_1 \frac{\alpha^{k_1+k_2}\Gamma(\alpha)}{\Gamma(\alpha+N)} \prod_{j=1}^{k_1}\Gamma(n_{1j}) \prod_{j=1}^{k_2}\Gamma(n_{2j}) \\
&+ (1-\pi_1) \frac{\alpha^{k_1+k_2}\Gamma(\alpha)^2}{\Gamma(\alpha+N_1)\Gamma(\alpha+N_2)} \prod_{j=1}^{k_1}\Gamma(n_{1j}) \prod_{j=1}^{k_2}\Gamma(n_{2j})
\end{aligned}
$$

else, if $k_0 > 0$:

$$
\begin{aligned}
&\Pi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1, \boldsymbol{q}_2) = \\
&\quad (1-\pi_1) \frac{\alpha^{k_1+k_2+k_0}\Gamma(\alpha)}{\Gamma(\alpha+N)} \prod_{j=1}^{k_1}\Gamma(n_{1j}) \prod_{j=1}^{k_2}\Gamma(n_{2j}) \prod_{j=1}^{k_0}\Gamma(q_{1j}+q_{2j})
\end{aligned}
$$

which can be rewritten down as in Camerlenghi et al. (2019); call

$$
\Phi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1 + \boldsymbol{q}_2) = \frac{\alpha^{k_1+k_2+k_0}\Gamma(\alpha)}{\Gamma(\alpha+N)} \prod_{j=1}^{k_1}\Gamma(n_{1j}) \prod_{j=1}^{k_2}\Gamma(n_{2j}) \prod_{j=1}^{k_0}\Gamma(q_{1j}+q_{2j})
$$

the EPPF of the fully exchangeable case, and

$$
\Phi_{k_0+k_i}^{(N_i)}(\boldsymbol{n}_i, \boldsymbol{q}_i) = \frac{\alpha^{k_i+k_0}\Gamma(\alpha)}{\Gamma(\alpha+N_i)} \prod_{j=1}^{k_i}\Gamma(n_{ij}) \prod_{j=1}^{k_0}\Gamma(q_{ij})
$$

the marginal EPPF for the individual groups $i = 1, 2$. We have that:

$$
\Pi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1, \boldsymbol{q}_2) = \pi_1 \Phi_k^{(N)}(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{q}_1 + \boldsymbol{q}_2) + (1-\pi_1)\Phi_{k_0+k_1}^{(N_1)}(\boldsymbol{n}_1, \boldsymbol{q}_1)\Phi_{k_0+k_1}^{(N_2)}(\boldsymbol{n}_2, \boldsymbol{q}_2) I(k_0 = 0)
$$

which is (6.13). □

Proof of **Proposition 6.3**.
Of course, the marginal law of $(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_I)$, conditional to $\boldsymbol{c}$, can be computed as

$$\mathcal{L}(d\boldsymbol{\theta}_1, \dots d\boldsymbol{\theta}_I \mid \boldsymbol{c}) = \int_{N_\Theta} \dots \int_{N_\Theta} \mathcal{L}(d\boldsymbol{\theta}_1, \dots d\boldsymbol{\theta}_I \mid \tilde{q}_1, \dots \tilde{q}_I, \boldsymbol{c}) \mathcal{L}(d\tilde{q}_1, \dots d\tilde{q}_I).$$

Now we operate a change of indices and call $\boldsymbol{\theta}_r = \{\boldsymbol{\theta}_i = (\theta_{i1}, \dots \theta_{iN_i}) : c_i = r\}$, so that $(\boldsymbol{\theta}_1, \dots \boldsymbol{\theta}_I) = (\boldsymbol{\theta}_{r_1}, \dots \boldsymbol{\theta}_{r_R})$ where R is the number of unique values in $\boldsymbol{c}$, i.e. the number of non-empty restaurants. We get

$$\begin{aligned}\mathcal{L}(d\boldsymbol{\theta}_1, \dots d\boldsymbol{\theta}_I \mid \boldsymbol{c}) &= \int_{N_\Theta} \int_{N_\Theta} \dots \int_{N_\Theta} \mathcal{L}(d\boldsymbol{\theta}_{r_1}, \dots d\boldsymbol{\theta}_{r_R} \mid \tilde{q}_1, \dots \tilde{q}_I, \boldsymbol{c}) \mathcal{L}(d\tilde{q}_1, \dots d\tilde{q}_I \mid \tilde{q}_0) \mathcal{L}(d\tilde{q}_0) \\ &= \int_{N_\Theta} \left(\prod_{i=1}^{R} \int_{N_\Theta} \mathcal{L}(d\boldsymbol{\theta}_{r_i} \mid \tilde{q}_{r_i}) \mathcal{L}(d\tilde{q}_{r_i} \mid \tilde{q}_0) \right) \mathcal{L}(d\tilde{q}_0).\end{aligned}$$

Observe that

$$\int_{N_\Theta} \mathcal{L}(d\boldsymbol{\theta}_{r_i} \mid \tilde{q}_{r_i}) \mathcal{L}(d\tilde{q}_{r_i} \mid \tilde{q}_0) = \mathcal{L}(\rho_{r_i}) \prod_{j=1}^{H_{r_i}} \widetilde{P}(d\theta^*_{r_i j}),$$

where ρ_{r_i} is the partition induced by the ℓ-clusters in the r_i restaurant. We use the same definition of ℓ-cluster as in Argiento et al. (2019). We underline that $\{\theta^*_{r_i j}, j = 1, \dots, H_{r_i}\}$ are not the unique values in the sample, since the base measure is atomic. Hence we have

$$\begin{aligned}&\int_{N_\Theta} \left(\prod_{i=1}^{R} \int_{N_\Theta} \mathcal{L}(d\boldsymbol{\theta}_{r_i} \mid \tilde{q}_{r_i}) \mathcal{L}(d\tilde{q}_{r_i} \mid \tilde{q}_0) \right) \mathcal{L}(d\tilde{q}_0) \\ &= \left(\prod_{i=1}^{R} \mathcal{L}(\rho_{r_i}) \right) \int_{N_\Theta} \prod_{i=1}^{R} \prod_{j=1}^{H_{r_i}} \tilde{p}(d\theta^*_{r_i j}) \mathcal{L}(d\tilde{q}_0).\end{aligned}$$

Now observe how the values $\{\theta^*_{rj} : \ r = 1, \dots R, j = 1, \dots H_{r_i}\}$ are all iid from $\tilde{p}$. So, there is no need for the division into restaurants anymore. We can thus stack all the vectors $\boldsymbol{\theta}^*_{r_i}$ together, apply a change of indices $(r_i, j) \to l$ so that now these $\{\theta^*_{r_i}\}$ are represented by $(\theta^*_1, \dots, \theta^*_L)$ and

$$\begin{aligned}\mathcal{L}(d\boldsymbol{\theta}_1, \dots d\boldsymbol{\theta}_I \mid \boldsymbol{c}) &= \prod_{i=1}^{R} \mathcal{L}(\rho_{r_i}) \int_{N_\Theta} \prod_{l=1}^{L} \tilde{p}(d\theta^*_l) \mathcal{L}(d\tilde{q}_0) \\ &= \prod_{i=1}^{R} \mathcal{L}(\rho_{r_i}) \int_{N_\Theta} \prod_{l=1}^{L} \left(\kappa G_0(d\theta^*_l) + (1-\kappa) \tilde{p}(d\theta^*_l)\right) \mathcal{L}(d\tilde{q}_0).\end{aligned}$$

Now, as done in Section 6.2.2, we introduce a set of latent variables $\boldsymbol{h} = (h_1, \dots, h_L)$, $h_l \overset{\text{iid}}{\sim} \text{Bernoulli}(\kappa)$, that gives

$$\begin{aligned}\mathcal{L}(d\boldsymbol{\theta}_1, \dots d\boldsymbol{\theta}_I \mid \boldsymbol{c}) &= \prod_{i=1}^{R} \mathcal{L}(\rho_{r_i}) \sum_{\boldsymbol{h} \in \{0,1\}^L} p(\boldsymbol{h}) \int_P \prod_{l=1}^{L} G_0(d\theta^*_l)^{h_l} \times \tilde{q}_0(d\theta^*_l)^{1-h_l} \mathcal{L}(d\tilde{q}_0) \\ &= \prod_{i=1}^{R} \mathcal{L}(\rho_{r_i}) \sum_{\boldsymbol{h} \in \{0,1\}^L} p(\boldsymbol{h}) \prod_{l=1}^{L} G_0(d\theta^*_l)^{h_l} \int_P \prod_{l=1}^{L} \tilde{q}_0(d\theta^*_l)^{1-h_l} \mathcal{L}(d\tilde{q}_0) \\ &= \prod_{i=1}^{R} \mathcal{L}(\rho_{r_i}) \sum_{\boldsymbol{h} \in \{0,1\}^L} p(\boldsymbol{h}) \prod_{l=1}^{L} G_0(d\theta^*_l)^{h_l} \times \mathcal{L}(\eta \mid \boldsymbol{h}) \prod_{k=1}^{M(\eta)} G_{00}(d\theta^{**}_k),\end{aligned}$$

where η is the partition of the $\{\theta_l^* : \ l = 1, \dots, L \text{ and } h_l = 0\}$, i.e. the partition of $\sum_{l=1}^L (1 - h_l)$ objects arising form the Dirichlet process $\tilde{q}_0$, while $\{\theta_k^{**}\}$ are the unique values among $\{\theta_l^* : \ l = 1, \dots, L \text{ and } h_l = 0\}$ and $p(\boldsymbol{h}) = \prod_{l=1}^L \kappa^{h_l}(1-\kappa)^{1-h_l}$ is the joint distribution of $\boldsymbol{h}$. □

Proof of **Proposition 6.4**.
Model M_2 defines a prior Π_2 on the space of densities $(p, q) \in N_{\mathbb{Y}} \times N_{\mathbb{Y}}$. On the other hand, model M_1 defines a prior on $N_{\mathbb{Y}}$. However, by embedding $N_{\mathbb{Y}}$ in the product space $N_{\mathbb{Y}} \times N_{\mathbb{Y}}$ via the mapping $p \mapsto (p, p)$, we can also consider the prior Π_1 induced by model M_1 as a measure on (a subset of) $N_{\mathbb{Y}} \times N_{\mathbb{Y}}$.

Now, showing that Π_2 satisfies the Kullback-Leibler property is a straightforward application of Theorem 3 in Wu and Ghosal (2008), under the same set of assumptions on the kernel $k(\cdot|\theta)$, and on p_0 and q_0, that we do not report here. Notice that these assumptions are satisfied when $k(\cdot|\theta)$ is the univariate Gaussian kernel with parameters given by the mean and the scale, and under standard regularity conditions on p_0 and q_0.

Now we turn our attention to Π_1. It is obvious to argue that Π_1 does not have the Kullback-Leibler property in the larger space $N_{\mathbb{Y}} \times N_{\mathbb{Y}}$, since it gives positive mass only to sets $\{(p, q) \in N_{\mathbb{Y}} \times N_{\mathbb{Y}} : p = q\}$. Consequently, if $p_0 \neq q_0$, one will have that for a small enough δ:

$$\Pi_1 \left((p, q) : D_{KL}((p, q), (p_0, q_0) < \delta \right) = 0,$$

thus proving that Π_1 does not have the Kullback-Leibler property.

In summary, under the same assumptions on p_0, q_0 and the kernel $k(\cdot \mid \theta)$ as in Ghosal et al. (2008), and assuming $p_0 \neq q_0$, we are comparing a model (M_2) with the Kullback-Leibler property against one (M_1) that does not have it. Theorem 1 in Walker et al. (2004) implies that the Bayes factor consistency is ensured. □

6.B Discussion of Bayes Factor consistency in the homogeneous case

When $p_0 = q_0$, consistency of the Bayes factor would require $BF_{12} \to +\infty$. This is a result we have not been able to prove so far, but it is worth pointing out the following relevant issues. To begin with, note that both models M_1 and M_2 have the Kullback-Leibler property. Several papers discuss this case, for example Corollary 3.1 in Ghosal et al. (2008), Section 5 in Chib and Kuffner (2016) and Corollary 3 in Chatterjee et al. (2020) in the general setting of dependent data. For more specific applications, refer also to Tokdar and Martin (2019) where the focus is on testing Gaussianity of the data under a Dirichlet process mixture alternative, Mcvinish et al. (2009) for goodness of fit tests using mixtures of triangular distribution and Bhattacharya and Dunson (2012) for data distributed over non-euclidean manifolds.

As pointed out in Tokdar and Martin (2019), the hypotheses in Corollary 3.1 by Ghosal et al. (2008) are usually difficult to prove, since they require a lower bound on the prior mass Π_2 around neighborhoods of $(p_0, p_0) \in \mathbb{P}_{\mathbb{Y}} \times \mathbb{P}_{\mathbb{Y}}$. To the best of our knowledge, this kind of bounds have been derived only for the very special kind of mixtures in Mcvinish et al. (2009). Similarly, the approach by Chib and Kuffner (2016) would require a knowledge of such lower bounds too (see for instance their Assumption 3). Corollary 3 in Chatterjee et al. (2020) does not apply in our case as well, because one of their main assumptions presumes that both models specify a population distribution (i.e. a likelihood) with density w.r.t some *common* σ–finite measure, together with the true distribution of the data. In our case M_1 specifies random probability measures that are absolutely continuous w.r.t the Lebesgue measure on $\mathbb{R}$, while under model M_2 the random probability measures have density under the Lebesgue measure on $\mathbb{R}^2$.

6.C Relabeling step

In the following, we adopt a slightly different notation to simplify the pseudocode notation. Figure 6.C.1 depicts the state at a particular iteration. We denote by ψ_{rh} the atoms in restaurant r arising from G_0 and with τ_h the atoms arising from G_{00}. Observe how in restaurant 1 the value τ_2 appears more than once.

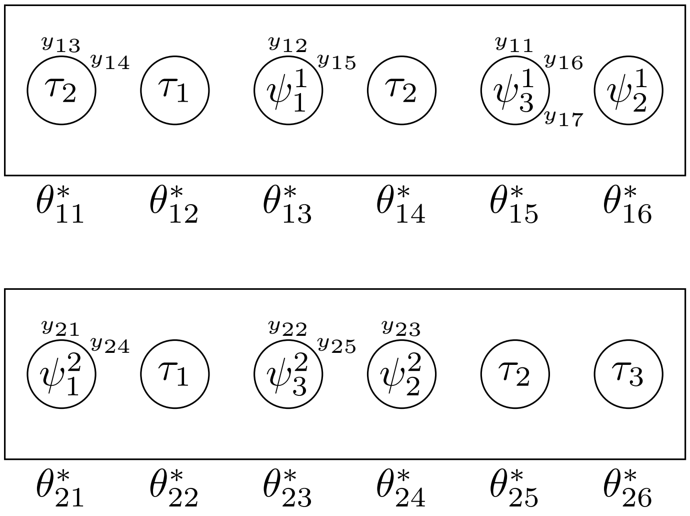


Figure 6.C.1: The state at one particular iteration

In our implementation, the state composed by $\boldsymbol{\psi}_r$, $\boldsymbol{\tau}$ (i.e. all the unique values of the atoms) and the indicator variables $\{t_{rl}\}$ and $\{h_{rl}\}$ that let us reconstruct the value of θ^*_{rl}. In particular if $\theta_{rl} = \psi_{rk}$ if $h_{rl} = 1$ and $t_{rl} = k$. Instead $\theta_{rl} = \tau_m$ if $h_{rl} = 0$ and $t_{rl} = m$. Moreover we also have the latent variables s_{ij} as described in Equation (6.16).

For the example in Figure 6.C.1, the latent variables assume the following values for the first restaurant

$$\boldsymbol{s}_1 = [5, 3, 1, 1, 3, 5, 5] \qquad \boldsymbol{h}_1 = [0, 0, 1, 1, 0, 0] \qquad \boldsymbol{t}_1 = [2, 1, 1, 3, 3, 3]$$

while for the second restaurant

$$\boldsymbol{s}_2 = [1, 3, 4, 1, 3] \qquad \boldsymbol{h}_2 = [1, 0, 1, 1, 0, 0] \qquad \boldsymbol{t}_2 = [1, 1, 3, 2, 2, 3]$$

During the relabeling step, we look at the number of customers in each table and find out that $\theta^*_{12}, \theta^*_{14}, \theta^*_{16}, \theta^*_{22}, \theta^*_{25}$ and θ^*_{26} are not used. Moreover also τ_1, τ_3 and ψ_2^1 are not used.

This leads to the following relabel

$$\boldsymbol{s}_1^{new} = [3, 2, 1, 1, 2, 3, 3] \qquad \boldsymbol{h}_1^{new} = [0, 1, 1] \qquad \boldsymbol{t}_1^{new} = [1, 1, 2]$$

and

$$\boldsymbol{s}_2^{new} = [1, 2, 3, 1, 2] \qquad \boldsymbol{h}_2^{new} = [1, 1, 1] \qquad \boldsymbol{t}_2^{new} = [1, 3, 2]$$

In the code, the transformation $\boldsymbol{s}_i \to \boldsymbol{s}_i^{new}$ is straightforward. Moreover $\boldsymbol{h}_i^{new}$ is computed from $\boldsymbol{h}_i$ by selecting only the elements corresponding to the sorted unique values in $\boldsymbol{s}_i$. For example the unique values in $\boldsymbol{s}_i$ are $[1, 3, 5]$ and $\boldsymbol{h}_i^{new} = [\boldsymbol{h}_i[1], \boldsymbol{h}_i[3], \boldsymbol{h}_i[5]]$.

The only complicated step is the one concerning $\boldsymbol{t}$. To update this last set of indicator variables we build two maps: τ_{map} and ψ_{map} that associate to the old labels the new ones. For example, we have that

$$\tau_{map} = \{2 \to 1\}$$
$$\psi_{map} = \{(1,3) \to (1,2)\}$$

meaning that all the τ_2's will be relabeled τ_1 and that ψ_3^1 will be relabeled ψ_2^1.

6.D Additional details on the simulation studies

6.D.1 A note on the mixing

One aspect of the inference presented so far that is clear from all the simulated scenarios, is that the posterior simulation of $\boldsymbol{c}$, and hence of the partition $\boldsymbol{\rho}$, usually stabilizes around one particular value and then very rarely moves. This could be interpreted as a mixing issue of the MCMC chain. However, notice that once the 'true' partition of the population is identified, it is extremely unlikely to move from that state, which can be seen directly from Equation (6.19). Indeed, moving from one state to another modifies the likelihood of an entire population. In particular, moving from a state where $c_i = c_j$ for two populations i and j that are *actually* homogeneous, to a state where $c_i \neq c_j$ is an extremely unlikely move.

To further illustrate the point, consider for ease of explanation the case of Simulation Scenario I where both populations are the same, and suppose that at a certain MCMC iteration we impute $c_1 = c_2 = 1$. In order for the chain to jump to $c_2 = 2$, the 'empty' mixing distribution $\tilde{q}_2$ must be sampled in such a way to give a reasonably high likelihood to all the data from the second population $y_{21}, \dots y_{2N_2}$; again, see (6.19). If one did not make use of pseudopriors, this would mean that $\tilde{q}_2$ would be sampled from the prior, thus making this transition virtually impossible. But even using pseudopriors, the transition remains quite unlikely. Indeed, once $c_1 = c_2 = 1$, we get an estimate of $\tilde{q}_1$ using data from the two homogeneous groups, hence getting a much better estimate that one would get when $c_1 \neq c_2$.

Nevertheless, in all simulation scenarios we tried this problem has not prevented the posterior simulation algorithm from identifying the correct partition of populations, as defined in these scenarios. In particular, we found that $P(\boldsymbol{\rho}_4^{true}|data) = 0.75$ only in scenario IV , while in all the other cases we tried, the values of $P(\boldsymbol{\rho}_4^{true}|data)$ was greater than 0.9. We also computed the cluster estimate of the posterior of $\boldsymbol{\rho}$ that minimizes the posterior expectation of Binder's loss (Binder, 1978) under equal misclassification costs and of the variation of information loss (Wade et al., 2018). In all the examples proposed, the 'true' partition was correctly detected by both estimates.

6.D.2 Additional Formulas and Plots

- Figure 6.D.2 reports the density estimates for Scenario I and Scenario III of the simulation study.

- Figure 6.D.1 reports the scatterplot of $P(c_1 = c_2 \,|\, data)$ (estimated through the MCMC samples) for the last simulation with 2 populations

- **Expression of Equation** (6.21).

Let $p_r = \sum_{i=1}^{H_r} w_{ri} \mathcal{N}(\mu_{ri}, \sigma_{ri}^2)$ and $p_m = \sum_{j=1}^{H_m} w_{mj} \mathcal{N}(\mu_{mj}, \sigma_{mj}^2)$ be the mixture densities associated to the mixing measures $\tilde{q}_r$ and $\tilde{q}_m$ respectively. Observe that both H_m and H_r are finite here as $\tilde{q}_r$ and $\tilde{q}_m$ have been approximated as shown in the description of the Gibbs sampler in Section 6.4. Then

$$\begin{aligned} d^2(\tilde{q}_r, \tilde{q}_m) = L_2^2(p_r, p_m) &= \int (p_r(y) - p_m(y))^2 dy \\ &= \int \left(\sum_{i=1}^{H_r} w_{ri} \mathcal{N}(y; \mu_{ri}, \sigma_{ri}^2) - \sum_{j=1}^{H_m} w_{mj} \mathcal{N}(y; \mu_{mj}, \sigma_{mj}^2) \right)^2 dy \end{aligned}$$

For any value of y the above integrand reduces to

$$\begin{aligned} &\left(\sum_{i=1}^{H_r} w_{ri} \mathcal{N}(y; \mu_{ri}, \sigma_{ri}^2) \right)^2 + \left(\sum_{j=1}^{H_m} w_{mj} \mathcal{N}(y; \mu_{mj}, \sigma_{mj}^2) \right)^2 + \\ &\qquad - 2 \left(\sum_{i=1}^{H_r} w_{ri} \mathcal{N}(y; \mu_{ri}, \sigma_{ri}^2) \right) \left(\sum_{j=1}^{H_m} w_{mj} \mathcal{N}(y; \mu_{mj}, \sigma_{mj}^2) \right) \end{aligned}$$

Each term in the right hand side can be expressed as a product of two summations, say $(\sum_i a_i)(\sum_j b_j) = \sum_{i,j} a_i b_j$. When $\{a_i\}$ and $\{b_j\}$ are equal, this further reduces to $\sum_{i,i'} a_i a_{i'}$.

Hence, exchanging summations and integrals, $d^2(\tilde{q}_r, \tilde{q}_m)$ equals

$$\begin{aligned} d^2(\tilde{q}_r, \tilde{q}_m) = &\sum_{i,i'=1}^{H_r} w_{ri}, w_{ri'} \int \mathcal{N}(y; \mu_{ri}, \sigma_{ri}^2) \mathcal{N}(y; \mu_{ri'}, \sigma_{ri'}^2) dy \\ &+ \sum_{j,j'=1}^{H_m} w_{mj}, w_{mj'} \int \mathcal{N}(y; \mu_{mj}, \sigma_{mj}^2) \mathcal{N}(y; \mu_{mj'}, \sigma_{mj'}^2) dy \\ &- 2 \sum_{i=1}^{H_r} \sum_{j=1}^{H_m} w_{ri} w_{mj} \int \mathcal{N}(y; \mu_{ri}, \sigma_{ri}^2) \mathcal{N}(y; \mu_{mj}, \sigma_{mj}^2) dy. \end{aligned}$$

As an immediate consequence of Equation (371) in Petersen and Pedersen (2012), we also have that all the integrals involved have a nice closed-form expression

$$\int \mathcal{N}(y; \mu, \sigma^2) \mathcal{N}(y; \mu', (\sigma')^2) dy = \mathcal{N}(\mu; \mu', \sigma^2 + (\sigma')^2).$$

Hence, $d^2(\tilde{q}_r, \tilde{q}_m)$ can be easily computed analytically.

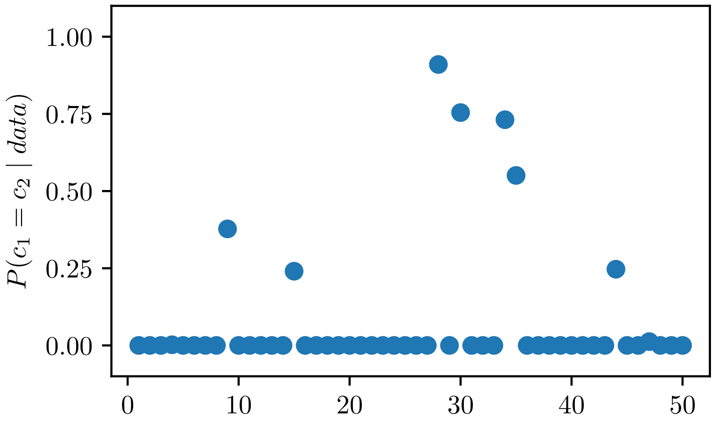


Figure 6.D.1: Plot of the posterior probabilities $P(c_1 = c_2|data)$ for all of the 50 simulated datasets.

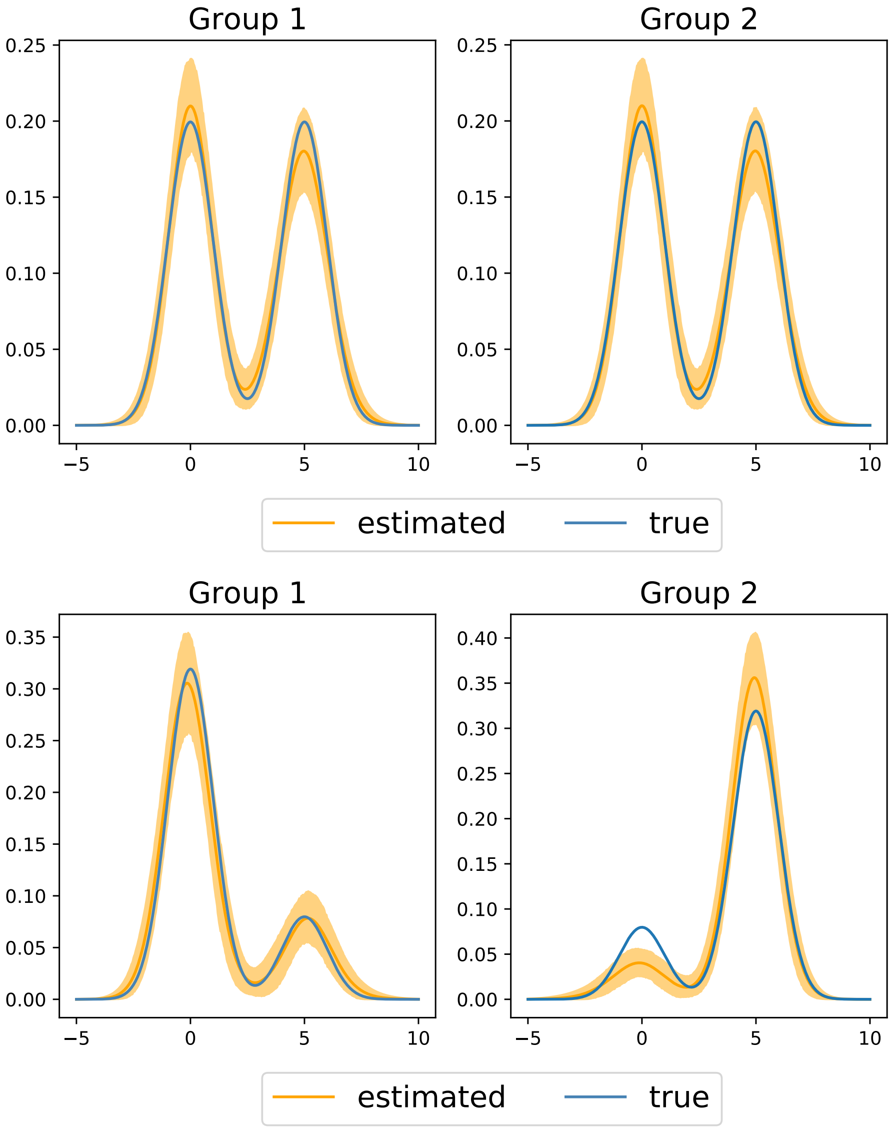


Figure 6.D.2: Density estimates and pointwise 95% posterior credible intervals for the two populations of Scenario I (top) and Scenario III (bottom).

7. Spatially dependent mixture models via the LogisticMCAR distribution

In this chapter, based on Beraha et al. (2021), we consider the problem of spatially dependent areal data, where for each area independent observations are available, and propose to model the density of each area through a finite mixture of Gaussian distributions. The spatial dependence is introduced via a novel joint distribution for a collection of vectors in the simplex, that we term logisticMCAR. We show that salient features of the logisticMCAR distribution can be described analytically, and that a suitable augmentation scheme based on the Pólya-Gamma identity allows to derive an efficient Markov Chain Monte Carlo algorithm. When compared to competitors, our model has proved to better estimate densities in different (disconnected) areal locations when they have different characteristics. We discuss an application on a real dataset of Airbnb listings in the city of Amsterdam, also showing how to easily incorporate for additional covariate information in the model.

7.1 Introduction

In spatial statistics, it is often assumed that data in neighboring locations are likely to behave more similarly than those that are far away. Thus, inference and prediction methods have been developed to take into account spatial dependence. Spatial data are classified into three main categories, according to Cressie (1992): geostatistical data, for which an exact location is known for each observation, areal (or lattice) data, when each observation is associated to a specific area or node in a lattice, and point patterns, where the object of the inference is the event location. Examples of the first are environmental applications (see Webster and Oliver, 2007) and geological reservoir characterization for oil and gas recovery (see Pyrcz and Deutsch, 2014, for examples). A recent review paper on statistical models for areal data is Banerjee (2016), which focuses on disease mapping and spatial survival analysis. Point patterns are often employed in ecology, as described in Velázquez et al. (2016). See also the textbook by Banerjee et al. (2014) for data classification, applications and statistical models and techniques for spatially dependent data.

7.1.1 Setup

We focus on areal data, and, in particular, we consider the problem of modeling data from I different groups, where each group corresponds to a specific areal location. More in detail, we assume that the spatial domain Ω is divided into I areas and, for each area, there is a vector of observations $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iN_i})$ from the same variable, each value y_{ij} corresponding to a different subject j in area i. The goal of this manuscript is the proposal of a statistical model, for data $\{\boldsymbol{y}_i, i = 1, \ldots, I\}$, accounting for dependence arising from spatial proximity while being flexible enough to model data that do not fit standard parametric distributions. We further assume that data, within each areal unit i, are independent and identically distributed (i.i.d.) from an area-specific density f_i; the problem we address is the joint estimation of spatially dependent densities $f_1, \ldots, f_I$.

We take the Bayesian viewpoint and we specify a prior for dependent densities $(f_1, \dots, f_I)$ that encourages distributions associated to areas that are spatially close to be more similar than those associated to areas that are far away. Relaxing the assumption of identically distributed observations within each area is straightforward in the regression context, i.e. when covariates for each subject are available.

As motivating application, we consider publicly available data on Airbnb listings in the city of Amsterdam (NL). Airbnb is the largest vacation rental marketplace. In recent years it has been debated that Airbnb has deeply transformed the social structure of major touristic cities, as Amsterdam (Van Der Zee, 2016), Barcelona (Garcia-Ayllon, 2018) and several US cities (Wachsmuth and Weisler, 2018), driving up property prices and disrupting communities. The application dataset consists of more than $17,000$ listings spread over neighborhoods in Amsterdam. Our goal is to predict the nightly price of a new listing, with information given by covariates, taking into account the spatial dependence. Such a model can be of interest to a 'new' lessor wishing to rent their house or flat on Airbnb. The area-specific estimate of the density might allow the lessor to understand the full market of renting apartments in his/her neighborhood, unlike a simple point estimate of the average price. The lessor might also understand if it is worth making home improvements in order to get a higher rent or assessing, for instance, the posterior predictive probability of the rent being above some threshold.

A peculiar feature of the municipality of Amsterdam is that three neighborhoods are not connected to the rest of the city but among themselves (see, for instance, Figure 7.7.1), i.e. there are two different connected components in the adjacency graph of neighborhoods. It is likely that the nightly prices exhibit substantially different behavior when comparing one component to the other. Hence, we want to build a model that encourages sharing of information across neighboring areas, but does not force densities belonging to different components to be similar a priori.

Compared to more traditional spatial regression techniques such as eigenvector spatial filtering (see Griffith et al., 2019, for a review), geographically weighted regression (Brunsdon et al., 1998) or the models in the `R` package *CARBayes* (Lee, 2013a), our approach does not make distributional assumptions (such as assuming Gaussian-distributed responses) and our focus here is on density modeling and estimation and density regression via mixture models.

7.1.2 Previous work on Bayesian spatial density modeling

To model our distributions we resort to the well established class of mixture models (Fruhwirth-Schnatter et al., 2019), that are a classical tool for density estimation. In the Bayesian nonparametric setting, since MacEachern (2000), a great effort has been dedicated to modeling a set of related, though not identical, distributions. Dealing with spatial processes, Gelfand et al. (2005) and Duan et al. (2007) developed a spatial dependent Dirichlet process as random-effects distribution in the context of point-reference data. The stick-breaking representation of the Dirichlet process allows all the models built from it to be considered as infinite mixture models. Starting from the stick-breaking representation of the dependent Dirichlet process in the particular case of *single atoms* (atoms not indexed by covariates), Dunson and Park (2008) proposed the kernel stick-breaking process mixtures; spatial extensions of these type of mixtures have been developed to accommodate for general covariates and spatial locations for geostatistical data, such as, e.g., Rodríguez and Dunson (2011) and Ren et al. (2011). Jo et al. (2017) considered mixture models based on species sampling priors where the spatial dependence is introduced through a Gaussian multivariate conditional autoregressive (CAR, Besag, 1974) model on a suitable transformation of the weights. Despite their focus being on point-referenced data, their model can be easily extended to areal data, as we do in Sections 7.4.3 and 7.6

for a comparison with our approach. The idea of building spatial dependence in mixture models through a CAR distribution on latent variables is also shared by Li et al. (2015), where the authors propose an area-dependent Dirichlet process that can also formally identify boundaries between areas, and Zhou et al. (2015), that use the trick of normalization of CAR distributions to time-varying weights in a rather complex application with focus on estimation of ambulance demand.

Despite the theoretical properties of Bayesian nonparametric mixtures, computing the posterior inference in this setting may yield computational issues. In fact, typical MCMC algorithms here would need to marginalize out the infinite dimensional distribution from the joint distribution of data and parameters, which might not be possible for models exhibiting a complicate dependence structure such as those mentioned above. As an alternative, finite-dimensional approximations of the infinite mixture representation are typically used in the MCMC algorithms. However, as recently pointed out by Lijoi et al. (2020b), the truncation procedure, for some models, might yield unwanted assumptions on the prior distribution of the number of clusters.

7.1.3 Our contribution and outline

In this chapter, we consider a finite mixture model, where the number H of components is fixed. Finite mixtures are particularly suited for the problem of modeling areal densities because (i) they adapt capturing the spatial dependence more than nonparametric mixtures, mainly because the weights of the finite mixtures are not forced to decrease exponentially fast to 0 as in many Bayesian nonparametric mixture models, and (ii) posterior inference under finite mixtures is extremely simple and admits efficient parallel code (unlike nonparametric models), thus helping our model scaling up as the size of the dataset increases. See Frühwirth-Schnatter (2006) and Celeux et al. (2019) for more insights on finite mixtures.

The first contribution of this work is the introduction of a joint distribution for a collection $\boldsymbol{w}_1, \dots, \boldsymbol{w}_I$ of I vectors in the simplex S^H, reflecting the areal proximity structure in the distribution, through a logistic transformation of Gaussian multivariate CAR models. This distribution has been termed here the logistic MCAR distribution. Other authors have considered similar tricks, e.g. Jo et al. (2017), who build on the CAR model by Clayton and Kaldor (1987).

A second contribution of this work is the proposal of a finite Gaussian mixture model for each of the I area-related densities, keeping in mind the flexibility of the Gaussian mixtures to accurately approximate smooth densities. We let all the mixtures share the same set of atoms, while introducing similarity between the different mixtures through the logistic MCAR distribution, that we use as a prior for the weights of the mixtures. Through simulated data examples and the Airbnb application we show how specific features of the proposed model include (i) a sparse mixture specification as meant in Malsiner-Walli et al. (2016) and (ii) densities corresponding to areal units which belong to two different connected components in the proximity graph may behave differently. We discuss this last particular point in our data illustrations.

A third contribution of this chapter is that we show how the full conditionals of the mixture weights can be sampled using a Gibbs sampler based on the Pólya-Gamma distribution, without resorting to Metropolis-Hastings steps, by exploiting a data augmentation scheme. As discussed in Polson et al. (2013), this update can lead to major improvements in the mixing of the chain. Our examples focus on continuous responses and the Gaussian kernel, though extensions to different kernels can be straightforwardly accommodated in our framework.

The rest of this article is organized as follows. Section 7.2 gives background on finite mixture models and the geometry on the finite-dimensional simplex. Section 7.3 illustrates the definition and properties of the joint distribution of a collection of I vectors in the simplex, taking into account the underlying spatial proximity matrix. Our area-dependent mixture model is illustrated in Section 7.4.1, and the sparse mixture specification is detailed in Section 7.4.2; Section 7.4.3 discusses on the differences between our spatial prior and that in Jo et al. (2017). Section 7.5 sketches the Gibbs sampler to compute the posterior and Section 7.6 presents results from two simulation studies with comparison with competitor models. The application to Airbnb Amsterdam is discussed in Section 7.7, where we propose two generalizations of our area-dependent mixture model to include subject-specific covariates and relaxing the identity in distribution assumption within each area. We conclude in Section 7.8 with final comments and discussion. The Appendix collects the proofs for the theoretical results, Monte Carlo simulations from the joint distribution of the I vectors in the simplex, full description of the Gibbs sampler, as well as additional plots and tables for the examples. Codes of our MCMC algorithm for simulated data and Airbnb Amsterdam application has been implemented `C++` and `Python` and is available at https://github.com/mberaha/spatial_mixtures.

7.2 Preliminaries

7.2.1 Mixture Models

For any areal unit $i = 1, \dots, I$ and subject $j = 1, \dots, N_i$, we assume observation $y_{ij} \in \mathbb{Y} \subset \mathbb{R}^p$. In this chapter, we fix $p = 1$, but multivariate responses can be straightforwardly accommodated in our context. A flexible model for the density in each area can be constructed by assuming a finite mixture, specifically

$$y_{ij} \mid \boldsymbol{w}_i, \boldsymbol{\tau}_i \overset{\text{iid}}{\sim} f_i(\cdot) = \sum_{h=1}^{H} w_{ih} k(\cdot \mid \tau_{ih}) \quad j = 1, \dots, N_i \tag{7.1}$$

where $k(\cdot \mid \tau)$ is a density on $\mathbb{Y}$ for any $\tau \in \Theta$, and Θ is the parameter space. Each vector $\boldsymbol{w}_i = (w_{i1}, \dots, w_{iH})^T$, the weights of the mixture (7.1), belongs to the $H - 1$ dimensional simplex S^H, where

$$S^H := \{(z_1, \dots, z_H) \in \mathbb{R}^H : 0 \le z_h \le 1, h = 1, \dots, H, \sum_{h=1}^{H} z_h = 1\} \tag{7.2}$$

and $\boldsymbol{\tau}_i = (\tau_{i1}, \dots, \tau_{iH})^T$ are parameters in Θ^H. In this chapter, we refer to $\boldsymbol{\tau}_i$ and $\boldsymbol{w}_i$ as the *atoms* and the *weights* of the mixture f_i.

Our goal is to introduce dependence between mixtures such that data in neighboring areas are more likely to be modeled with similar distributions than data in far areal units. A general mixture model like (7.1) would require to model jointly both the atoms and the weights of all the mixtures, in order to obtain a dependence structure suitable for spatial applications, which can be a challenging task in general, unless we consider a very specific application. In our approach instead, borrowing ideas from the single atom dependent Dirichlet processes, we constrain all the atoms across the different areas to be equal, i.e. $\boldsymbol{\tau}_1 = \boldsymbol{\tau}_2, \ldots = \boldsymbol{\tau}_I = \boldsymbol{\tau}$, and focus only on the weights of the mixtures. In this way, a sufficient condition for two different mixtures to be similar is to have similar weights. In general, it is more difficult to define mixtures with area-dependent weights than generalizing to area-dependent weights and atoms, since simulation algorithms for models based on standard mixture models can usually be adapted with few modifications to dependent atoms.

When the goal of the inference is cluster estimation, the choice of H might become crucial. An alternative consists in assuming H random, including it in the state space of the MCMC algorithm; see, for instance, Nobile (1994). However, inference in this setting can be computationally intensive as it needs to rely either on specifically designed trans-dimensional MCMC moves (see Green, 1995; Richardson and Green, 1997), or to numerically evaluate infinite series, as in Miller and Harrison (2018) and in the marginal sampler in Argiento and De Iorio (2022). On the other hand, sparse mixture models, as meant in Malsiner-Walli et al. (2016), assume a large value for H, larger than needed, and a prior assigning large mass to configurations where the weights of the superfluous components assume values close to zero. This implies that the prior number of *non-empty* components (i.e. components where at least one observation is allocated to) is significantly smaller than H.

In Section 7.3 we propose a prior distribution for $(\boldsymbol{w}_1, \dots, \boldsymbol{w}_I)$, in such a way that weights associated to close areas are more similar than weights associated to areas farther away, by constructing a Markov random field for random vectors with bounded sum. Moreover, by assuming a prior on the hyperparameters, we also show that this prior can induce sparsity in the mixture (see Section 7.4.2) as in Malsiner-Walli et al. (2016).

7.2.2 Geometry on the simplex S^H

The simplex $S^H \subset \mathbb{R}^H$ defined in (7.2) is not a vector subspace of $\mathbb{R}^H$. However, S^H is a vector space when equipped with the so-called Aitchison geometry, that defines the operation of *perturbation* (analogous of addition), *powering* (analogous of multiplication by scalar) and *inner product.* If $\boldsymbol{w}, \boldsymbol{w}_1, \boldsymbol{w}_2 \in S^H$, $\alpha \in \mathbb{R}$ we have

$$\boldsymbol{w}_1 \oplus \boldsymbol{w}_2 = \mathcal{C}(w_{11}w_{21}, \dots, w_{1H}w_{2H}) := \left(\frac{w_{11}w_{21}}{\sum_{i=1}^H w_{1i}w_{2i}} \cdots \frac{w_{1H}w_{2H}}{\sum_{i=1}^H w_{1i}w_{2i}} \right)$$

$$\alpha \odot \boldsymbol{w} = \mathcal{C}(w_1^\alpha, \dots, w_H^\alpha) \qquad\qquad \langle \boldsymbol{w}_1, \boldsymbol{w}_2 \rangle = \frac{1}{2H} \sum_{i,j=1}^H \log \frac{w_{1i}}{w_{1j}} \log \frac{w_{2i}}{w_{2j}}$$

where $\mathcal{C}$ denotes the *closure*, or normalization (i.e. dividing each element by the sum of all the elements) of a vector in $\mathbb{R}^H$. The symbols $\oplus$, $\odot$ and $\langle \cdot, \cdot \rangle$ denote perturbation, powering and inner product, respectively.

Many maps from S^H to $\mathbb{R}^{H-1}$ are available in the literature. For our purpose we focus on the bijective additive log-ratio transformation (alr), defined by alr : $\boldsymbol{w} \mapsto \tilde{\boldsymbol{w}}$:

$$\tilde{w}_j = \log \frac{w_j}{w_H}, \quad j = 1, \dots, H-1$$

and its inverse, $\boldsymbol{w} = \text{alr}^{-1}(\tilde{\boldsymbol{w}}) := \mathcal{C}(\mathrm{e}^{\tilde{w}_1}, \dots, \mathrm{e}^{\tilde{w}_{H-1}}, 1)$, that is

$$w_j = \frac{\mathrm{e}^{\tilde{w}_j}}{1 + \sum_{h=1}^{H-1} \mathrm{e}^{\tilde{w}_h}}, \; j = 1, \dots, H-1, \quad w_H = 1 - \sum_{h=1}^{H-1} w_h = \frac{1}{1 + \sum_{h=1}^{H-1} \mathrm{e}^{\tilde{w}_h}} \; . \tag{7.3}$$

Observe that both maps are linear, i.e., for any $\boldsymbol{w}_1, \boldsymbol{w}_2 \in S^H$, $\tilde{\boldsymbol{w}}_1, \tilde{\boldsymbol{w}}_2 \in \mathbb{R}^{H-1}$, $\alpha \in \mathbb{R}$,

$$\text{alr}(\boldsymbol{w}_1 \oplus \boldsymbol{w}_2) = \text{alr}(\boldsymbol{w}_1) + \text{alr}(\boldsymbol{w}_2), \quad \text{alr}(\alpha \odot \boldsymbol{w}_1) = \alpha \; \text{alr}(\boldsymbol{w}_1)$$
$$\text{alr}^{-1}(\tilde{\boldsymbol{w}}_1 + \tilde{\boldsymbol{w}}_2) = \text{alr}^{-1}(\tilde{\boldsymbol{w}}_1) + \text{alr}^{-1}(\tilde{\boldsymbol{w}}_2), \quad \text{alr}^{-1}(\alpha \tilde{\boldsymbol{w}}_1) = \alpha \odot \text{alr}^{-1}(\tilde{\boldsymbol{w}}_1).$$

The alr transformation is often applied in the context of compositional data analysis, where statistical inference for data in the simplex has been pioneered by Aitchison (1986). In particular, this map was used in Aitchison and Shen (1980) to define a new distribution on the simplex, the logistic-normal distribution. Formally, we say that $\boldsymbol{w} = (w_1, \dots, w_{H-1},$

$w_H := 1 - \sum_{h=1}^{H-1} w_h)^T \in S^H$ follows the logistic-normal distribution of parameters $\boldsymbol{\mu}, \Sigma$ for $\boldsymbol{\mu} \in \mathbb{R}^{H-1}$, and Σ a positive definite $(H-1) \times (H-1)$ matrix if

$$\tilde{\boldsymbol{w}} = \text{alr}(\boldsymbol{w}) = \left(\log \frac{w_1}{w_H}, \ldots, \log \frac{w_{H-1}}{w_H}\right)^T \sim \mathcal{N}_{H-1}(\boldsymbol{\mu}, \Sigma)$$

where $\mathcal{N}_{H-1}(\boldsymbol{\mu}, \Sigma)$ denotes the $(H-1)$-dimensional Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ. The logistic-normal distribution offers a rich way to model data embedded on the simplex and is particularly suited for our application. Although moments of this distribution exist, their expression is not available analytically. However, when modeling data in the simplex, one is usually more interested in the pairwise ratios of the components than on the values of the components themselves. In turn, these expected values and covariances are available analytically and given by

$$\mathbb{E}\left[\log \frac{w_i}{w_j}\right] = \mu_i - \mu_j, \quad \text{Cov}\left(\log \frac{w_i}{w_j}, \log \frac{w_l}{w_k}\right) = \Sigma_{il} + \Sigma_{jk} - \Sigma_{ik} - \Sigma_{jl}$$

where Σ_{il} denotes the (i, l)-element of the matrix Σ.

7.3 The logistic MCAR distribution

In this section, we introduce and describe a joint distribution for a collection of vectors in the simplex $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_I \in S^H$, reflecting the areal proximity structure in the distribution. For each pair of areas i and j, $g_{ij} \in [0, 1]$ indicates the amount of spatial proximity between them. In the rest of the chapter we assume $g_{ij} = 1$ if i and j are neighbors, i.e. the areas share at least a border, and $g_{ij} = 0$ otherwise, but we could consider more general settings. By definition, $g_{ii} = 0$ for all i. The matrix $G = [g_{ij}]_{i,j=1}^I$ is called the proximity matrix and we assume it known. It will be useful, for our analyses, to identify the matrix G with a graph, whose nodes are denoted by indexes $1, \ldots, I$ and the links are given by the g_{ij}'s, i.e. there is a link between nodes i and j if, and only if, $g_{ij} = 1$. We define the joint distribution of $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_I$ introducing the transformed vectors $\tilde{\boldsymbol{w}}_i := \text{alr}(\boldsymbol{w}_i)$, $i = 1, \ldots, I$ and assuming a joint Gaussian conditional autoregressive distribution for $(\tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_I)$.

Conditionally autoregressive (CAR) models are a special case of Markov random fields. In general, if $\{X_1, \ldots, X_n\}$, with $X_i \in \mathbb{R}$, is a set of random variables, to define a CAR model over $X_1, \ldots, X_n$, one usually starts by assigning the conditional distribution of each X_i given all the others $X_{-i} := \{X_1, \ldots, X_{i-1}, X_{i+1}, \ldots X_n\}$. The set of conditional distributions, under assumptions, identifies the unique joint distribution of $(X_1, \ldots, X_n)$. The class of CAR models is large; see further detail in Besag (1974), Cressie (1992), Cressie (1993), Kaiser and Cressie (2000), Cressie and Wikle (2015) and references therein, just to include a few papers.

We generalize the univariate CAR model in Leroux et al. (2000) assuming the following multivariate conditionally autoregressive (MCAR) model:

$$\tilde{\boldsymbol{w}}_i \,|\, \tilde{\boldsymbol{w}}_{-i}, \Sigma, \rho \sim \mathcal{N}_{H-1}\left(\frac{\rho \sum_{j=1}^I g_{ij} \tilde{\boldsymbol{w}}_j + (1-\rho)\tilde{\boldsymbol{m}}_i}{\rho \sum_{j=1}^I g_{ij} + 1 - \rho}, \frac{\Sigma}{\rho \sum_{j=1}^I g_{ij} + 1 - \rho}\right), \tag{7.4}$$

where $i = 1, \ldots, I,$, Σ is a definite positive $(H-1) \times (H-1)$ matrix and $\tilde{\boldsymbol{m}}_i \in \mathbb{R}^{H-1}$ for all i. When $H - 1 = 1$, (7.4) gives the prior proposed in Leroux et al. (2000).

Denoting with $A \otimes B$ the Kronecker product between the $m \times n$ matrix A and the $p \times q$ matrix B , i.e., $A \otimes B$ is the $pm \times qn$ matrix with entries $(A \otimes B)_{pr+v,qs+w} = A_{r,s}B_{v,w}$, the following proposition guarantees that the joint distribution of $\tilde{\boldsymbol{w}} = vec(\tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_I)$, the vectorization of the weights, is well defined and unique.

Proposition 7.1. *Assume that $\rho \in (-1,1)$ and $\tilde{\boldsymbol{m}}_i = \tilde{\boldsymbol{m}}_j$ if areas i and j belong to the same connected component of the graph G. Then the set of full conditionals in (7.4) defines a unique joint probability distribution for $\tilde{\boldsymbol{w}} = vec(\tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_I)$, given by*

$$\tilde{\boldsymbol{w}} \sim \mathcal{N}_{I(H-1)}\left(\tilde{\boldsymbol{m}}, \left((F - \rho G) \otimes \Sigma^{-1}\right)^{-1}\right) \tag{7.5}$$

where $\tilde{\boldsymbol{m}} = vec(\tilde{\boldsymbol{m}}_1, \ldots, \tilde{\boldsymbol{m}}_I)$ and $F = diag(\rho \sum_j g_{1j} + 1 - \rho, \ldots, \rho \sum_j g_{Ij} + 1 - \rho)$.
Proof: see 7.A.

The matrix $A^{-1}(G,\rho) := (F - \rho G) = \rho\,(diag(G\mathbf{1}_I) - G) + (1-\rho)\mathbb{I}_I$ in (7.5), where $\mathbf{1}_I \in \mathbb{R}^I$ denotes the vector of ones and $\mathbb{I}_I$ denotes the $I \times I$ identity matrix, has a key role here. When $\rho = 1$, (7.4) reduces to the intrinsic CAR model, and the joint density of $(\tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_I)$ is improper. If $\rho = 0$, the $\tilde{\boldsymbol{w}}_i$'s are independent. See below for further properties of $A(G, \rho)$.

We say that the sequence of vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_I$ follows a logistic multivariate CAR distribution of parameters ρ and Σ on a graph G if the transformed variables $(\tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_I)$, $\tilde{\boldsymbol{w}}_i = \mathrm{alr}(\boldsymbol{w}_i)$, follow the MCAR model in (7.4) (or (7.5)). We write $(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_I) \sim$ logisticMCAR$(\tilde{\boldsymbol{m}},\, \rho, \Sigma; G)$.

One key aspect is the relation that (7.4) induces over the vectors on the simplex rather than on their alr-transformation. This is made clear by the following proposition.

Proposition 7.2. *If $(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_I) \sim logisticMCAR(\tilde{\boldsymbol{m}}, \rho, \Sigma; G)$, then, for any $i = 1, \ldots, I$,*

$$\mathbb{E}\left[\log \frac{w_{il}}{w_{ik}} \mid \boldsymbol{w}_{-i}\right] = \log\left(\left(\frac{m_{il}}{m_{ik}}\right)^{1-\rho} \prod_{j \in U_i} \left(\frac{w_{jl}}{w_{jk}}\right)^{\rho}\right)^{(\rho|U_i|+1-\rho)^{-1}} \qquad l,k = 1, \ldots, H \tag{7.6}$$

where $U_i = \{j : g_{ij} > 0\}$, $|U_i| = \sum_j g_{ij}$ and $\boldsymbol{m}_i = (m_{i1}, \ldots, m_{iH})$, with $\boldsymbol{m}_i = alr^{-1}(\tilde{\boldsymbol{m}}_i)$.
Proof: see 7.A.

There are several immediate but interesting properties of (7.6). First of all, if $\rho = 1$, (7.6) means that the expected value of (the logarithm of) the ratios between the components of $\boldsymbol{w}_i$ is equal to (the logarithm of) the geometric mean of the corresponding ratios of the components of the vectors $\boldsymbol{w}_j$ nearby. If $\rho = 0$, the right hand side of (7.6) reduces to $\log(m_{il}/m_{ik})$, which is to be expected since, in this case, the $\boldsymbol{w}_i$'s would not be spatially correlated. Instead, in case $0 < \rho < 1$, which we assume throughout the chapter (see Section 7.4.1), we can interpret the right hand side of (7.6) as a weighted mean on the simplex, according to Aitchison geometry, of two components: the first component m_{il}/m_{ik} corresponding to the mean $\boldsymbol{m}$ and the second $\prod_{j \in U_i}(w_{jl}/w_{jk})$ taking into account the spatial dependence. In other words, Proposition 7.2 provides the same interpretation of (7.4) but for ratios between components of the anti-transformed vectors in the simplex, if we look at them through the Aitchison geometry.

Starting from the joint distribution in (7.5), we can also study the marginal covariance of $\boldsymbol{w}_i, \boldsymbol{w}_j$ in $(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_I) \sim$ logisticMCAR$(\tilde{\boldsymbol{m}}, \rho, \Sigma; G)$ for $i \neq j$. We point out that the matrix $A(G, \rho)^{-1}$, introduced above, is a strictly diagonal dominant matrix (i.e. for each row, the absolute value of the diagonal entry is larger than or equal to the sum of the absolute values of the off-diagonal entries in that row) with negative off-diagonal entries, and, hence, its inverse $A(G, \rho)$ has elements which are all positive.

Proposition 7.3. *If $(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_I) \sim logisticMCAR(\tilde{\boldsymbol{m}}, \rho, \Sigma; G)$, then*

$$Cov\left(\log \frac{w_{il}}{w_{im}}, \log \frac{w_{jl}}{w_{jm}}\right) = A_{ij}\left(\Sigma_{ll} - 2\Sigma_{lm} + \Sigma_{mm}\right) \quad \forall i,j,\ l,m = 1, \ldots, H-1$$

$$Cov\left(\log \frac{w_{il}}{w_{iH}}, \log \frac{w_{jl}}{w_{jH}}\right) = A_{ij}\Sigma_{ll} \quad i,j = 1, \ldots, I \quad l = 1, \ldots, H-1$$

In particular, if areas i and j belong to different connected graph components of the graph G, $Cov\left(\log\frac{w_{il}}{w_{im}}, \log\frac{w_{jl}}{w_{im}}\right) = 0$ and $\tilde{\boldsymbol{w}}_i, \tilde{\boldsymbol{w}}_j$ are independent, conditioning to parameters $\tilde{\boldsymbol{m}}, \rho, \Sigma$.
Proof: see 7.A.

Observe that the logisticMCAR distribution of $(\boldsymbol{w}_1, \dots, \boldsymbol{w}_I)$ is not exchangeable, i.e. it is not true that $\mathcal{L}(\boldsymbol{w}_1, \dots, \boldsymbol{w}_I)$ and $\mathcal{L}(\boldsymbol{w}_{\pi(\{1\})}, \dots, \boldsymbol{w}_{\pi(\{I\})})$ are equal for any $(\pi(\{1\}), \dots, \pi(\{I\}))$ permutation of $(1, \dots, I)$. Here, as in the rest of the chapter, the distribution of a random element y is denoted by $\mathcal{L}(y)$. Nonetheless, the logisticMCAR distribution induces exchangeable priors on all the fully connected components of the graph G.

The logisticMCAR distribution shares the same limitation as the logistic-normal one, i.e. moments are not available in closed-form expressions. In 7.B we report an extensive Monte Carlo (MC) simulation where we compute the covariance between different components of the vectors of weights and we draw a comparison between the logisticMCAR and the Dirichlet distributions.

7.4 Spatially dependent mixture models

We return to the problem of formalizing a Bayesian model for I groups of data $(\boldsymbol{y}_1, \dots, \boldsymbol{y}_I)$, $\boldsymbol{y}_i = (y_{i1}, \dots, y_{iN_i})$, $i = 1, \dots, I$. As mentioned at the beginning of Section 7.3, we assume that each vector $\boldsymbol{y}_i$ is associated to an area i and that for each pair of areas i and j, $g_{ij} = 1$ if i and j are neighbors and $g_{ij} = 0$ otherwise.

7.4.1 The finite mixture model with spatially dependent weights

Let the proximity matrix $G = [g_{ij}]_{i,j=1}^{I}$ be fixed. We assume that $\boldsymbol{y}_1, \dots, \boldsymbol{y}_I$, conditioning to $\boldsymbol{w}_1, \dots, \boldsymbol{w}_I$ and $\boldsymbol{\tau}$, are independent and that, for each $i = 1, \dots, I$,

$$y_{ij} \mid \boldsymbol{w}_i, \boldsymbol{\tau} \overset{\text{iid}}{\sim} \sum_{h=1}^{H} w_{ih} \mathcal{N}(\cdot \mid \tau_h) \quad j = 1, \dots, N_i, \tag{7.7}$$

$$\tau_h \overset{\text{iid}}{\sim} P_0 \quad h = 1, \dots, H \tag{7.8}$$

$$(\boldsymbol{w}_1, \dots, \boldsymbol{w}_I) \mid \rho, \Sigma \sim \text{logisticMCAR}(\tilde{\boldsymbol{m}}, \rho, \Sigma; G) \tag{7.9}$$

$$\Sigma \sim \text{Inv-Wishart}(\nu, V) \tag{7.10}$$

$$\rho \sim \pi(\rho) \tag{7.11}$$

where notation $\overset{\text{iid}}{\sim}$ denotes independent and identically distributed random variables, $\boldsymbol{w}_i = (w_{i1}, \dots, w_{iH})^T \in S^{H-1}$ (see (7.2)) and $\tilde{\boldsymbol{m}} = vec(\tilde{\boldsymbol{m}}_1, \dots, \tilde{\boldsymbol{m}}_I) \in \mathbb{R}^{I(H-1)}$. As often considered, we study the case where the kernel in the mixture (7.7) is the Gaussian density with mean μ_h and variance σ_h^2, so that $\tau_h = (\mu_h, \sigma_h^2)$ and P_0 is a probability distribution over $\Theta = \mathbb{R} \times \mathbb{R}^+$. Specific choices of P_0 are discussed in Sections 7.6 and 7.7. We consider independent marginal priors for ρ and Σ. Moreover, the support of the prior of ρ is typically assumed to be $(0, 1)$ to induce the similarity of spatial neighbors (see, for instance, Gelfand and Vounatsou, 2003, Section 4).

Model (7.7) - (7.11) assumes that each group of data $\boldsymbol{y}_i$ is modeled as a (finite) mixture of Gaussian kernels. Specifically, observations within each group are i.i.d given the weights and the atoms of the mixtures, while conditionally to all the mixture weights $\boldsymbol{w}_1, \dots, \boldsymbol{w}_I$, observations in different groups are independent. All the I mixtures share the same set of atoms $\tau_1, \dots, \tau_H$, which are assumed i.i.d from the base measure P_0, a continuous distribution on $\Theta = \mathbb{R} \times \mathbb{R}^+$. The dependence between mixtures in different areal units is induced only by the prior on the mixture weights. In order to derive a Gibbs sampler for our model, we introduce the latent variables s_{ij}, one for each observation, indicating to which component of the mixture observations are allocated to, and rewrite (7.7) as

$$y_{ij} \mid s_{ij} = h, \tau_h \overset{\text{ind}}{\sim} \mathcal{N}(\cdot \mid \tau_h) \quad j = 1, \ldots, N_i, \; i = 1, \ldots, I \tag{7.12}$$

$$p(s_{ij} = h \mid \boldsymbol{w}_i) = w_{ih} \quad h = 1, \ldots, H \tag{7.13}$$

where notation $\overset{\text{ind}}{\sim}$ is used to represent independent random variables (from different distributions). A component in the mixture is said *empty* if it has not been allocated to any observation. Here, and in the whole chapter, *cluster* denotes any allocated component and the number of clusters is the number of allocated components. It is clear from (7.12)-(7.13) that the allocated and empty components, as well as the number of clusters, are random variables, with marginal prior distributions induced by our model.

We complete the specification of our model by adopting a marginal prior on $\tilde{\boldsymbol{m}}$ that encourages sparsity in the mixtures. We discuss this choice in detail in the next Section 7.4.2.

7.4.2 Sparse mixtures via a prior on $\tilde{m}$

Generally, a sparse mixture is obtained when the number of clusters is smaller than the total number of components H. There are two well-known strategies to obtain sparse mixtures in the Bayesian context. The first one assigns a prior on the weights that forces them to be stochastically decreasing, so that the 'last' weights are very small and the corresponding mixture components are seldom allocated. The alternative strategy consists in assigning a prior for the weights that concentrates its mass around the edges of the simplex in a symmetric way, as it is the case of the sparse Dirichlet distribution, i.e. a Dirichlet distribution with all the parameters equal to α, with $0 < \alpha < 1$. In the latter case, there is no preferential ordering of the weights and any mixture component could be allocated. We think that the first approach might not fit spatial applications, in particular when the proximity graph G has disconnected components, since assuming decreasing weights for all the mixtures would force data from two disconnected components to be always sampled from the few components with larger weights, and hence to behave always similarly.

Here, we show how we can mimic the sparse Dirichlet distribution for the weights, by assuming a suitable prior on parameters $\tilde{\boldsymbol{m}}_i$'s in our model. We start by observing that in the mixture model (7.7) for the i-th area, if coordinate values in the vector $\tilde{\boldsymbol{m}}_i$ in (7.9) are very different among each other, this would force some components h in (7.7) to be more often allocated than others, being their weights larger than the others (in mean). Hence, we induce 'symmetric' sparsity in our marginal prior for the weights by assuming $\tilde{\boldsymbol{m}}_i \sim \mathcal{N}_{H-1}(\boldsymbol{0}, \eta^2 \mathbb{I})$. Observe that, since the distribution of $\tilde{\boldsymbol{m}}_i$ is centered in $\boldsymbol{0}$ and isotropic, we are not forcing, marginally, any specific ordering on the weights.

To understand the role of η^2, let us consider an illustrative example when $H = 3$ and $I = 1$. Let $\boldsymbol{m}_1 = \text{alr}^{-1} \tilde{\boldsymbol{m}}_1$ and consider $d_{12} = (\log(m_{11}/m_{1H}) - \log(m_{12}/m_{1H}))^2$, which corresponds to the distance between m_{11} and m_{12} in the Aitchison geometry. We may consider d_{12} as a plug-in estimator of the distance between w_{11} and w_{12}. The largest values of d_{12} are obtained when one between m_{11}, m_{12} and m_{1H} is approximately 1 and the others are close to zero. Moreover, from $\tilde{\boldsymbol{m}}_1 \sim \mathcal{N}_2(\boldsymbol{0}, \eta^2 \mathbb{I})$, we have that $d_{12}/(2\eta^2)$ has chi-squared distribution with one degree of freedom. Hence the random variable d_{12} is stochastically increasing with η^2.

This feature holds also for larger values of H as shown in Figure 7.4.2, where the behavior of $\boldsymbol{w}_i$, for different values of η^2, is illustrated. We conclude that η^2 is a sparsity tuning parameter and sparsity of the $\boldsymbol{w}_i$'s is obtained for larger values of η^2. Note that this is the opposite behavior of other sparsity priors, such as the double exponential or the horseshoe (Bhadra et al., 2019), where a distribution with significant mass near zero is assumed. Because our parameters are transformed through the logistic map (7.3), assuming a prior concentrated in zero for $\tilde{\boldsymbol{m}}$ would result in a prior concentrated on $(1/H, \dots, 1/H)$ for $\boldsymbol{w}$, of course this being far from sparsity.

Moreover, observe that Proposition 7.1 requires

$$\{\tilde{\boldsymbol{m}}_i = \tilde{\boldsymbol{m}}_j = \tilde{\boldsymbol{m}}_{C_m} \text{ if } i, j \in C_m \text{ for some } m\} \tag{7.14}$$

where $C_1, \dots, C_k$ denote the connected components of graph G, i.e. all the parameters $\tilde{\boldsymbol{m}}_i$s are assumed common within each connected component. This condition is obviously met if all the $\tilde{\boldsymbol{m}}_i$'s are equal. However, this seems overly restrictive, since we would loose the property that two connected graph components in $(\tilde{\boldsymbol{w}}_1, \dots, \tilde{\boldsymbol{w}}_I)$ are independent under CAR distributions when marginalizing out the only shared parameter $\tilde{\boldsymbol{m}}_1$. Hence, we propose to extend the logisticMCAR$(\tilde{\boldsymbol{m}}, \rho, \Sigma; G)$ in (7.9) assuming

$$\tilde{\boldsymbol{m}}_{C_1}, \dots \tilde{\boldsymbol{m}}_{C_k} \overset{\text{iid}}{\sim} \mathcal{N}_{H-1}(\boldsymbol{0}, \eta^2 \mathbb{I}). \tag{7.15}$$

7.4.3 Comparison with competitor models

As mentioned in the Introduction, we have defined a prior for $(\boldsymbol{w}_1, \dots, \boldsymbol{w}_I)$, allowing weights associated to close areas to be more similar than weights associated to areas farther away, through the logistic transformation of a Gaussian CAR model. The idea is not new in the literature, and the prior for the mixture weights of area-dependent densities in Jo et al. (2017) is closely related to our prior. We discuss the differences between the two priors in this section and we further compare their features by fitting simulated data to the two models in Section 7.6.1.

We briefly introduce the class of spatially dependent species sampling mixtures in Jo et al. (2017), who define the weights in the mixtures to be spatially dependent, modeling them from a Gaussian CAR distribution, as we do. Their focus is on geo-referenced data (with multiple observations in each geographic location), and they propose two different CAR specifications, namely the Mercer CAR and the Clayton-Kaldor CAR (Clayton and Kaldor, 1987) priors. Since it is not straightforward to extend the Mercer CAR formulation to areal data as it requires the computation of a geographical distance rather than defining a proximity matrix, we only consider the Clayton-Kaldor CAR species sampling model in Jo et al. (2017) for comparison. We have shown in Section 7.4.2 that our marginal prior can mimic the sparse Dirichlet distribution by assuming $\tilde{\boldsymbol{m}}$ in the logisticMCAR$(\tilde{\boldsymbol{m}}, \rho, \Sigma; G)$ to be random. Below, we discuss how sparsity is obtained also in the spatially dependent species sampling model in Jo et al. (2017), but only in some sort of 'asymmetric' manner, and how this impacts the modeling of different connected components in the graph G.

In the following, we refer to the prior in Jo et al. (2017) as *CK-SSM*. Instead of jointly modeling the transformed weights in each location, Jo et al. (2017) assume independent univariate CAR model for (a transformation of) the weights associated to each component of the mixture in the different areas. Recall that they assume a mixture with infinite components, i.e., $h = 1, 2, \dots$. With our notation, let $\boldsymbol{\nu}_h = (w_{1h}, \dots, w_{Ih})$, then the CK-SSM prior for $\boldsymbol{\nu}_h$ is

$$\widetilde{\boldsymbol{\nu}}_h \overset{\text{ind}}{\sim} \mathcal{N}_I(\widetilde{\theta}_h, \tau^2(I - \rho G)) \quad h = 1, 2 \dots, \quad \nu_{ih} = w_{ih} = \frac{e^{\widetilde{\nu}_{hi}}}{\sum_j e^{\widetilde{\nu}_{ji}}} \quad i = 1, \dots, I \tag{7.16}$$

In order to guarantee that the denominator of the fraction in (7.16) is finite, Jo et al. (2017) assume that $\widetilde{\theta}_h$ is a vector with all components equal to $\log\{1-(1+e^{b-ah})^{-1}\}$, a and b being positive hyperparameters. In force of that, the weights $\widetilde{\nu}_{hi}$ are stochastically decreasing with h for each area i. This ordering is preserved by the exponential and normalization transformations, so that w_{ih} will be stochastically decreasing with h as well, for each i. Note also that (7.16) makes $\boldsymbol{w}$ non-identifiable.

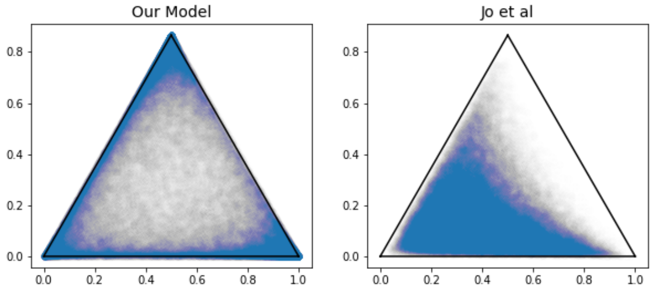
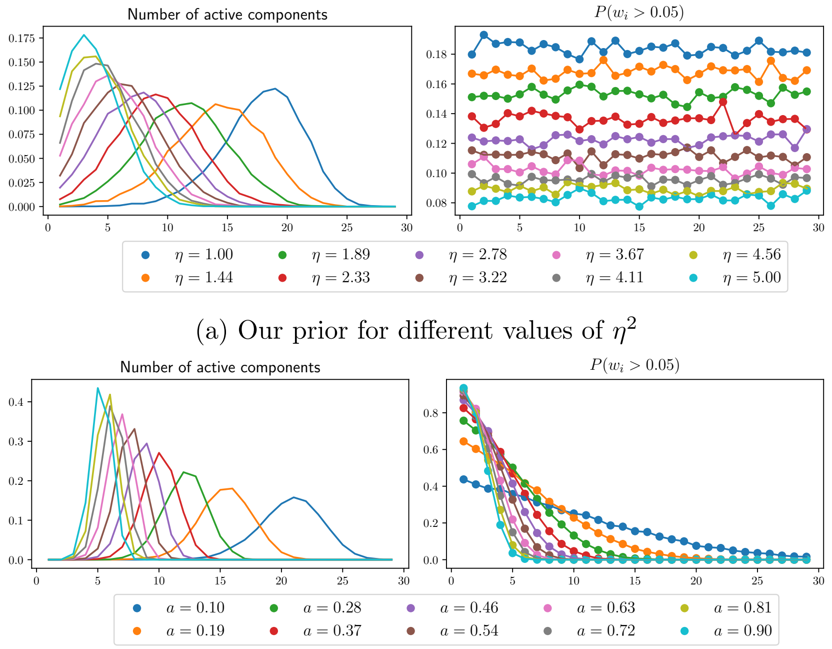


Figure 7.4.1: Scatterplots of $N = 100,000$ MC draws (in blue) from the marginal priors of the weights in one single area with $H = 3$, under our logisticMCAR prior (left) and (7.16) in Jo et al. (2017) (right). White/gray areas represents low-density zones and dark blue zones high-density ones.

We start by considering $I = 1$ area and drop the subscript i. As in Jo et al. (2017), we truncate (7.16) to the first H terms for computation. Figure 7.4.1 shows a comparison of the marginal priors of the weights in the mixture (7.7) with $H = 3$, under our logisticMCAR prior and (7.16) introduced in Jo et al. (2017). In particular, for our logisticMCAR($\tilde{\boldsymbol{m}}, \rho, \Sigma; G$) prior we have assumed $\tilde{\boldsymbol{m}} \sim \mathcal{N}_2(\boldsymbol{0}, 9\mathbb{I})$, $\tilde{\boldsymbol{w}} \sim \mathcal{N}_2(\tilde{\boldsymbol{m}}, \mathbb{I})$, that is (7.5) with $\Sigma = \mathbb{I}$, while we fix $\widetilde{\theta}_h = \log\{1 - (1 + \mathrm{e}^{1-h})^{-1}\}$, $h = 1, \dots, H$, in (7.16) as in Jo et al. (2017) ($a = b = 1$) and $\tau^2 = 1$.

We compare the priors via $N = 100,000$ MC draws. Figure 7.4.1 shows the scatterplots of the draws of the two marginal priors. In particular, the draws from our prior (left panel) recover the 'sparse' symmetric Dirichlet prior with all parameters equal to $\alpha < 1$; the draws are symmetrically concentrated around the edges of the simplex, and give significant mass to locations near the vertexes. On the other hand, the draws from the CK-SSM prior clearly show asymmetry in favor of the first component, also giving negligible mass to neighborhoods of the vertices. When the number H of components in the mixture (7.7) is larger, we can compare the priors via two functionals by computing (i) the number of active components ($H^{(a)}$), that we define as the components associated to weights greater than 0.01, i.e. the cardinality of the set $\{h : \ w_h > 0.01\}$, and (ii) the probability for each component of the vector $\boldsymbol{w} \in S^H$ to be greater than the threshold 0.05. We fix $H = 30$ and, simulating $N = 10,000$ MC draws as before, we plot the marginal prior distributions of these functionals under our logisticMCAR prior (Figure (7.4.2a)) and (7.16) in Jo et al. (2017) (Figure (7.4.2b)), for different values of the hyperparameters in the priors. From both left panels, displaying the marginal priors of $H^{(a)}$ (as continuous lines to help seeing the differences), it is clear that the two models may induce different types of prior behaviors. However, when considering the right panels, displaying, for each index $h = 1, \dots, H$, the probability that $w_h > 0.05$, it is clear that, while under our prior, for each degree of sparsity η^2, the probability of inclusion of a single component does not show a preferential ordering, this probability decreases with h under the CK-SSM prior. Going back to the prior in (7.15), observe how this model specification gives a major difference with the mixture model in Jo et al. (2017).

This is particularly relevant if we aim at considering the context where areal units are connected through the graph G, but there are at least two different connected components, as we will have in the application in Section 7.7. Intuitively from the discussion above, CK-SSM would still force the different connected components in the graph to behave similarly, because of the parameter $\widetilde{\boldsymbol{\theta}}$ shared by all the mixtures; see (7.16). We tested this scenario more in detail by considering a spatial domain subdivided in four areas with



(a) Our prior for different values of η^2

(b) The prior in Jo et al. (2017) for different values of a and $b = 0.5$

Figure 7.4.2: Prior distribution of the number of active components (left) and the probability for w_h to be greater than 0.05 (right) under our prior (top row) and prior (7.16) in Jo et al. (2017) (bottom row). Here $H = 30$.

two connected components $\{1, 2\}, \{3, 4\}$. Figure 7.D.1 shows the total variation distance for $(\boldsymbol{w}_1, \boldsymbol{w}_2)$ and $(\boldsymbol{w}_1, \boldsymbol{w}_4)$ under the logisticMCAR and CK-SSM priors, having fixed hyperparameters as above and $\rho = 0.95$. It is clear that, as sparsity increases, the distance between $\boldsymbol{w}_1$ and $\boldsymbol{w}_4$ increases under the logisticMCAR but decreases under the CK-SSM, showing how imposing a sparse behavior in the CK-SSM prior forces similar distributions in disconnected components of the graph. See also Figure 7.D.2 for a visual representation of draws from the prior distributions. This effect becomes more and more evident as the sparsity in each mixture is increased, as shown in Figure 7.4.2b. On the other hand, our model allows for the required level of sparsity in each mixture without forcing the different connected components in the graph to behave similarly. For this reason, we believe we have introduced a more flexible model for jointly estimate spatially dependent densities than Jo et al. (2017), at least for applications where different connected components in the graph should exhibit different behaviors.

We will provide comparison also with the Hierarchical Dirichlet Process (HDP) mixture model in Teh et al. (2006) in Section 7.6. To keep the chapter self-contained as much as possible, we report the HDP mixture model as follows

$$y_{ij} \mid F_i \overset{\text{iid}}{\sim} \int_\Theta k(y_{ij} \mid \tau) F_i(d\tau), \quad \{F_i\}_{i=1}^I \mid G \overset{\text{iid}}{\sim} \mathcal{D}_{\alpha G} \quad G \sim \mathcal{D}_{\beta P_0} \tag{7.17}$$

where $\mathcal{D}_{\beta P_0}$ denotes the Dirichlet measure, i.e. the distribution of a random probability measure that is the Dirichlet process with measure parameter βP_0. We assume the kernel $k(\cdot \mid \tau)$ as the Gaussian density on $\mathbb{Y}$ for $\tau = (\mu, \sigma^2)$ as in (7.7). Thanks to the stick-breaking representation of the Dirichlet process, it is possible to rewrite the likelihood in (7.17) as

$$y_{ij} \mid \{w_{ih}\}_{h=1}^\infty, \{\phi^*_{ih}\}_{h=1}^\infty \overset{\text{iid}}{\sim} \sum_{h=1}^\infty w_{ih} k(y_{ij} \mid \phi^*_{ih})$$

where $\phi^*_{ih} \mid G \overset{\text{iid}}{\sim} G$ in (7.17) and $\{w_{ih}\}$ for each i are a sequence of non-negative weights summing to 1. Moreover, since each F_i, conditionally to G, is an independent draw from the Dirichlet process prior with discrete base measure G, this yields that all the atoms are shared across all populations. This means that the set of the unique values in $\{\phi^*_{ih}\}_{h=1}^{\infty}$ is equal to the set of unique values in $\{\phi^*_{jh}\}_{h=1}^{\infty}$ for $j \neq i$ and coincides with the set of atoms in G. Hence, denoting by $\{\tau_h\}_{h=1}^{\infty}$ the atoms in G, the HDP mixture model defines a joint probability distribution for random probability measures with the same support points, as in our model. This is the motivation to consider the HDP mixture model as the 'natural competitor' of ours.

7.5 The Gibbs Sampler

We illustrate a MCMC algorithm to sample from the posterior distribution of our model (7.7)-(7.11) and (7.14) - (7.15). The state is described by parameters $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_H)$, $(\tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_I)$, where $\tilde{\boldsymbol{w}}_i = \operatorname{alr}(\boldsymbol{w}_i)$, $i = 1, \ldots, I$, $\{s_{ij}\}_{ij}$ $(j = 1, \ldots, N_i)$ in (7.12)-(7.13) and $\tilde{\boldsymbol{m}}_{C_1}, \ldots \tilde{\boldsymbol{m}}_{C_k}$ in (7.15).

We use the following notation: given the sequence of vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_I$, we denote by $\boldsymbol{w}_{-i}$ the same sequence where the i–th vector has been removed. Given a single vector $\boldsymbol{w}_i$, we denote by $\boldsymbol{w}_{i,-h}$ the same vector where the h-component as been removed. Finally, for the matrix Σ, let Σ_{ij} denote the (i,j)-element; moreover, Σ_i denotes its i–th row (as a vector) so that $\Sigma_{i,-j}$ denotes the i–th row where the j–th element has been removed and $\Sigma_{-h,-k}$ denotes the $(H-2) \times (H-2)$ matrix there the h–th row and k–th column have been removed.

There are two 'non-standard' steps in the Gibbs sampler: the update of the transformed weights $\tilde{\boldsymbol{w}}_i$ and the update of their means $\tilde{\boldsymbol{m}}_{C_1}, \ldots \tilde{\boldsymbol{m}}_{C_k}$. Here, we only describe the full conditionals of each $\tilde{\boldsymbol{w}}_i$. The full conditional of $\tilde{\boldsymbol{m}}_{C_i}$ is a multivariate Gaussian distribution. See 7.C for more detail on it, together with the other standard full conditionals.

We begin by writing the full conditional for $\tilde{w}_{ih}$, for each i and h, as

$$\mathcal{L}(\tilde{w}_{ih} \mid \tilde{\boldsymbol{w}}_{-i}, \tilde{\boldsymbol{w}}_{i,-h}, rest) \propto \pi(\tilde{w}_{ih} \mid \tilde{\boldsymbol{w}}_{-i}, \tilde{\boldsymbol{w}}_{i,-h}, \rho, \Sigma)\, \mathcal{L}(\tilde{w}_{ih} \mid \boldsymbol{s}_i, \tilde{\boldsymbol{w}}_{i,-h}) \tag{7.18}$$

where $\boldsymbol{s}_i = (s_{i1}, \ldots, s_{iH})^T$. The conditional prior $\pi(\tilde{w}_{ih} \mid \tilde{\boldsymbol{w}}_{-i}, \tilde{\boldsymbol{w}}_{i,-h}, \rho, \Sigma)$ can be derived from (7.5) conditioning with respect to the other components of the vector $\tilde{\boldsymbol{w}}_i$; we find

$$\pi(\tilde{w}_{ih} \mid \tilde{\boldsymbol{w}}_{-i}, \tilde{\boldsymbol{w}}_{i,-h}, \rho, \Sigma) = \mathcal{N}(\mu^*_{ih}, \Sigma^*_{ih}),$$

where $\mu^*_{ih} = \mu_{ih} + \Sigma_{h,-h}\Sigma^{-1}_{-h,-h}(\tilde{\boldsymbol{w}}_{i,-h} - \boldsymbol{\mu}_{i,-h})$ and $\Sigma^*_{ih} = (\rho\sum_{j=1}^{I} g_{ij} + 1 - \rho)^{-1}$ $\left(\Sigma_{h,h} - \Sigma_{h,-h}\Sigma^{-1}_{-h,-h}\Sigma_{-h,h}\right)$ by standard properties of the normal distribution, with $\boldsymbol{\mu}_i = (\rho\sum_{j=1}^{I} g_{ij} + 1 - \rho)^{-1}(\rho\sum_{j=1}^{I} g_{ij}\tilde{\boldsymbol{w}}_j + (1-\rho)\tilde{\boldsymbol{m}}_i)$. Moreover, using the same data augmentation scheme proposed in Holmes and Held (2006), we write the term $\mathcal{L}(\tilde{w}_{ih} \mid \boldsymbol{s}_i, \tilde{\boldsymbol{w}}_{i,-h})$ as

$$\mathcal{L}(\tilde{w}_{ih} \mid \boldsymbol{s}_i, \tilde{\boldsymbol{w}}_{i,-h}) = \left(\frac{\mathrm{e}^{\eta_{ih}}}{1 + \mathrm{e}^{\eta_{ih}}}\right)^{N_{ih}} \left(\frac{1}{1 + \mathrm{e}^{\eta_{ih}}}\right)^{N_i - N_{ih}}$$

where $\eta_{ih} = \tilde{w}_{ih} - C_{ih}$, $C_{ih} = \log\sum_{k \neq h} \mathrm{e}^{\tilde{w}_{ik}}$ (with $\tilde{w}_{iH} := 0$) and N_{ih} is the number of observations in area i assigned to component h.

To be able to sample from the full conditional of $\tilde{w}_{ih}$, we express $\mathcal{L}(\tilde{w}_{ih} \mid \boldsymbol{s}_i, \tilde{\boldsymbol{w}}_{i,-h})$ using an augmentation technique, based on the Pólya-Gamma distribution. The trick is analogous to that in Polson et al. (2013), in this case without covariates. We describe it in detail in the next paragraphs.

We denote by $\omega \sim PG(b,c)$ a random variable with a Pólya-Gamma distribution with parameters b and c, i.e.

$$\omega = \frac{1}{2\pi^2} \sum_{k=1}^{+\infty} \frac{g_k}{(k-1/2)^2 + c^2/(4\pi^2)} \tag{7.19}$$

where $g_k \overset{\text{iid}}{\sim} Gamma(b,1)$ and $b, c > 0$. The data-augmentation technique based on the Pólya-Gamma distribution relies on the following integral identity:

$$\frac{(\mathrm{e}^{\eta})^a}{(1+\mathrm{e}^{\eta})^b} = 2^{-b}\mathrm{e}^{(a-b/2)\eta} \int_0^{+\infty} \mathrm{e}^{-\omega\eta^2/2} p(\omega) d\omega$$

where $p(\omega)$ is the density of the $PG(b,0)$ random variable.

Taking advantage from the above equality, when introducing the latent variable $\omega_{ih} \sim PG(N_i, 0)$, we can derive the following full conditional for $\tilde{w}_{ih}$:

$$\mathcal{L}(\tilde{w}_{ih} \,|\, \tilde{\boldsymbol{w}}_{-i}, \tilde{\boldsymbol{w}}_{i,-h}, \boldsymbol{s}_i, \rho, \Sigma, \omega_{ih}) = N(\hat{\mu}_{ih}, \hat{\Sigma}_{ih}) \tag{7.20}$$

where

$$\hat{\mu}_{ih} = \left(\frac{\mu^*_{ih}}{\Sigma^*_h} + N_{ih} - N_i/2 + \omega_{ih} C_{ih} \right) \left(\frac{1}{\Sigma^*_{ih}} + \omega_{ih} \right)^{-1} \quad \hat{\Sigma}_{ih} = \left(\frac{1}{\Sigma^*_{ih}} + \omega_{ih} \right)^{-1}.$$

Moreover, the full conditional of ω_{ih} can be expressed as

$$\mathcal{L}(\omega_{ih} \,|\, \tilde{\boldsymbol{w}}_i) = PG\Big(N_i, \tilde{w}_{ih} - \log \sum_{k \neq h} e^{\tilde{w}_{ik}}\Big). \tag{7.21}$$

See 7.C for the proof of Equations (7.20)-(7.21).

These equations give a two steps Gibbs update for the variable $\tilde{w}_{ih}$. Indeed, one can first sample ω_{ih} from (7.21) (which depends on $\tilde{w}_{ih}$) and secondly update $\tilde{w}_{ih}$ from (7.20) (which depends on ω_{ih}). In this way, we are able to make two Gibbs steps in an augmented state space instead of a single Metropolis Hastings step. There are two reasons why one should prefer the former algorithm to the latter. First, the two-Gibbs-steps simulation avoids the choice of a proposal density for the update, that can be difficult due to the shape of the logistic transformation applied to the weights. Moreover, using the Pólya Gamma augmentation trick can be helpful in settings where the number of observations in a single area is not significantly greater than the number of components in the mixture, as we consider in Section 7.6.1, scenario II; see Section S6.3 of the supplementary material in Polson et al. (2013) for an explanation of this statement.

7.6 Simulated data

We consider two simulation studies to illustrate the flexibility of our model; in particular we will see that the model is able to exploit spatial dependence between densities corresponding to close areas. In the first example, we compare our model (SPMIX) with the Clayton-Kaldor Species Sampling Model of Jo et al. (2017) (CK-SSM) and the HDP mixture model (see (7.17)), that we use as a sort of black-box model for density estimation of grouped data. In the second example we generate data from spatially dependent densities and we check if our model is flexible enough to recover such dependence.

We run the Gibbs sampler for our model (7.7)-(7.11) together with the prior specification (7.14) - (7.15) (see Section 7.5 and 7.C), and the *direct sampler* for the HDP-mixture model in Teh et al. (2006). Both algorithms were coded in `C++`. In addition, we have

	Area	1	2	3	4	5	6
Scenario I	Density	$t(6,-4,1)$	$t(6,-4,1)$	$SN(4,4,1)$	$SN(4,4,1)$	$\chi^2(3,0,1)$	$\chi^2(3,0,1)$
	N_i	1000	1000	1000	1000	1000	1000
Scenario II	Density	$t(6,-4,1)$	$t(6,-4,1)$	$SN(4,4,1)$	$SN(4,4,1)$	$\chi^2(3,0,1)$	$\chi^2(3,0,1)$
	N_i	1000	10	1000	10	1000	10
Scenario III	Density	$t(6,-4,1)$	$t(6,-4,1)$	$SN(4,4,1)$	$SN(4,4,1)$	$Ca(0,1)$	$Ca(0,1)$
	N_i	100	100	100	100	100	100

Table 7.6.1: Non-Gaussian simulated data: true densities and sample sizes for each area under all scenarios

also implemented the CK-SSM model in `Stan` (Stan Development Team, 2018) with the prior (7.16). All the MCMC chains were run for 10,000 iterations after discarding the first 10,000 iterations as burn-in, keeping one every five iterations, resulting in a final sample size of 2,000, unless otherwise specified. In all cases, convergence was checked using both visual inspection of the chains and standard diagnostics available in the CODA package.

The base measures for our model, for the HDP-mixture and for the CK-SSM mixture are assumed all equal (and denoted by P_0) to match the models under comparison. Unless otherwise stated, we assume P_0 equal to the Normal-inverse-gamma distribution with parameters $\mu_0 = 0, a = b = 2, \lambda = 0.1$, i.e. $\mu \mid \sigma^2 \sim \mathcal{N}\left(\mu_0, \lambda^{-1}\sigma^2\right)$, $\sigma^2 \sim IG(a,b)$ and the prior in (7.11) as $\rho \sim Beta(1,1)$. For the HDP, the total mass parameters α and β are fixed and equal to 1. For our model, we set the prior hyperparameters for the marginal prior (7.10) of Σ as $\nu = 100$ and $V = \mathbb{I}$ for all the simulated examples. For the CK-SSM, we followed the hyperparameter tuning outlined in their paper, except for the parameters a and b that we fix to $a = 0.1$ and $b = 0.5$.

As metrics to compare the density estimates, i.e. the posterior mean of the density evaluated on a fixed grid, we use the Kullback-Leibler divergence and the Hellinger distance between the estimated density and the true one.

7.6.1 Non-Gaussian simulated data

We consider three scenarios. In each scenario we generate, for $I = 6$ different areas, an i.i.d. sample from a density that is not Gaussian: namely t-student (t), skew-normal (SN), chi-squared (χ^2) and Cauchy (Ca). The matrix G is fixed and represents a graph with only three connected components $\{1,2\}$, $\{3,4\}$, $\{5,6\}$. The three scenarios differ in the number of data in each area and in the data generating densities, as reported in Table 7.6.1: $t(\nu, \mu, \sigma)$ denotes the Student's t distribution with ν degrees of freedom, centered in μ and scaled by a factor σ; $SN(\xi, \omega, \alpha)$ denotes the Skew normal distribution with mean $\xi + \omega\alpha/\sqrt{1+\alpha^2}\sqrt{2/\pi}$, $\chi^2(k,0,1)$ denotes the standard chi-squared distribution with k degrees of freedom and $Ca(0,1)$ the Cauchy distribution. They cover extremely different cases: in Scenario I a large number of data is available in each area, so that borrowing strength from nearby areas would be superfluous; we actually expect our model to perform worse than the HDP-mixture, being the latter fully nonparametric. On the other hand, in Scenario II there are three areas (2, 4 and 6) with few data points (only 10). In this case, we expect our model to express its strength and give a better density estimate than the HDP-mixture, especially in those areas where few data are present. Finally, Scenario III is an in-between condition, where not so many observations as in Scenario I are available in each area. We also compare the results obtained with the CK-SSM mixture.

In order to make a fair comparison between our model, CK-SSM and the HDP models, we fixed the number of components H in our mixtures and in the CK-SSM to 10. This choice was made by looking at the posterior distribution of the number of components under the HDP-mixture in the different scenarios; we found that the number of clusters

	Model	1	2	3	4	5	6
Scenario I	SPMIX	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.02 ± 0.01	0.02 ± 0.01
	HDP	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.02 ± 0.01	0.02 ± 0.01
	CK-SSM	0.92 ± 0.46	0.92 ± 0.46	0.97 ± 0.16	0.98 ± 0.16	1.10 ± 0.31	1.10 ± 0.31
Scenario II	SPMIX	0.02 ± 0.00	0.04 ± 0.04	0.02 ± 0.01	0.02 ± 0.07	0.03 ± 0.01	0.03 ± 0.10
	HDP	0.01 ± 0.00	0.13 ± 0.04	0.03 ± 0.01	0.21 ± 0.07	0.03 ± 0.01	0.32 ± 0.10
	CK-SSM	0.91 ± 0.40	0.90 ± 0.40	0.97 ± 0.17	0.97 ± 0.17	1.22 ± 0.45	1.23 ± 0.44
Scenario III	SPMIX	0.15 ± 0.19	0.15 ± 0.18	0.09 ± 0.25	0.09 ± 0.25	0.06 ± 0.12	0.06 ± 0.12
	HDP	0.16 ± 0.19	0.16 ± 0.18	0.26 ± 0.25	0.26 ± 0.25	0.13 ± 0.12	0.13 ± 0.12
	CK-SSM	0.86 ± 0.33	0.86 ± 0.34	1.25 ± 0.29	1.25 ± 0.29	0.86 ± 0.41	0.86 ± 0.42

Table 7.6.2: Kullback-Leibler divergences between the true densities and the estimated ones, aggregated over 100 simulated datasets with ± one standard deviation

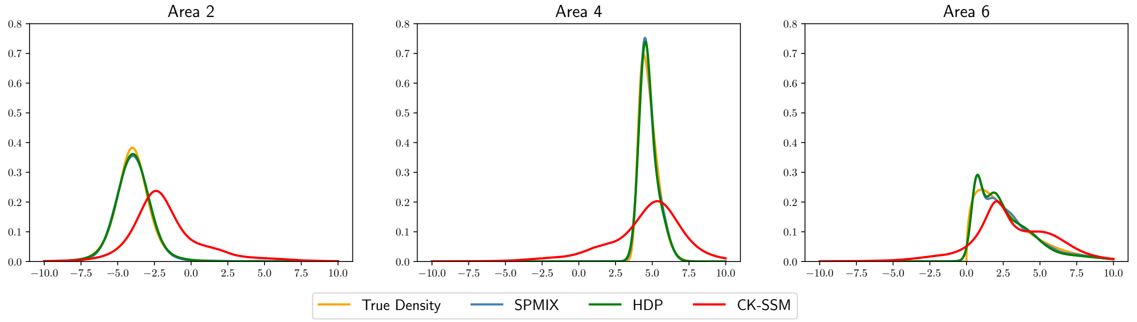


Figure 7.6.1: Non-Gaussian simulated data, Scenario I: true densities for areas 2, 4 and 6 and the corresponding density estimates under the three different models.

ranges between 3 and 10. For each scenario we repeatedly simulated 100 independent datasets. Table 7.6.2 shows the KL-divergence between the true density and the estimate under the three models. We average those values over the 100 simulated datasets, also considering ± one empirical standard deviation of the 100 values obtained. Table 7.D.1 in 7.D, reports the same values for the Hellinger distance. From both tables, we can see that in all the three scenarios, the CK-SSM has the worst performance in recovering the true data generating density. This reflects what we discussed in Section 7.4.3: the prior of such model forces mixture weights to be too similar across different connected components in the graph. We can clearly see this for example from Figure 7.6.1, where the density estimates for areas 4 and 6 (not connected in G) are close under the CK-SSM but not under our model. The HDP-mixture gives overall better estimates than those under our model in Scenario I. In this case, both density estimates are close enough to the true densities; see Figure 7.6.1. As we expect, under Scenario II, our model gives a better density estimate (than the HDP-mixture) in areas 2, 4 and 6, where only 10 data points are available; see Figure 7.6.2. Indeed, our model retrieves information from the neighboring areas, overcoming the lack of data in some of the areas. Interestingly, our model performs better in areas 3-6, and similarly in areas 1 and 2, under Scenario III, probably because of 'extreme' data in areas 5 and 6, where we generate data from a Cauchy distribution. This behavior is evident from Figure 7.6.2, being the 95% point-wise credible interval of the posterior distribution of the density much wider in HDP than in our approach. Finally, it is clear that our model fits data well also when the true density is highly non-symmetric, such as in locations 3-6 in Scenarios I and II.

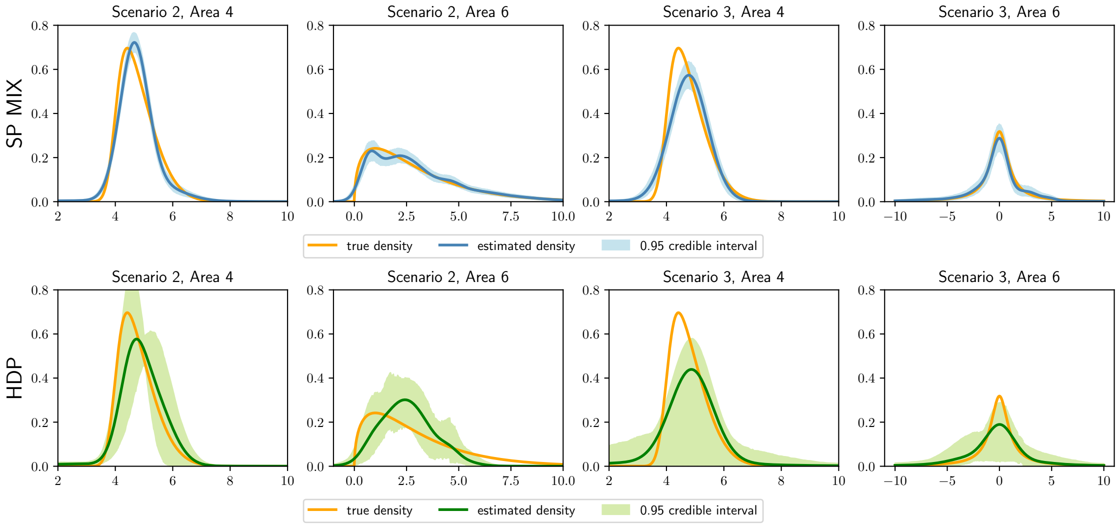


Figure 7.6.2: Non-Gaussian simulated data, Scenario II and III: estimated and true densities for areas 4 and 6 under our (top row) and HDP mixture (bottom row) models.

7.6.2 Simulation from spatially dependent weights

In the second simulation example we apply our model to estimate spatially dependent densities in contiguous areas, placed in a squared grid with a total number I of areas, in a unit area squared domain; we study three different scenarios, choosing $I = 16, 64, 256$. In the i-th area, we simulate observations as follows:

$$y_{ij} \overset{\text{iid}}{\sim} w_{i1}\mathcal{N}(-5,1) + w_{i2}\mathcal{N}(0,1) + w_{i3}\mathcal{N}(5,1) \quad j = 1,\dots,25 \tag{7.22}$$

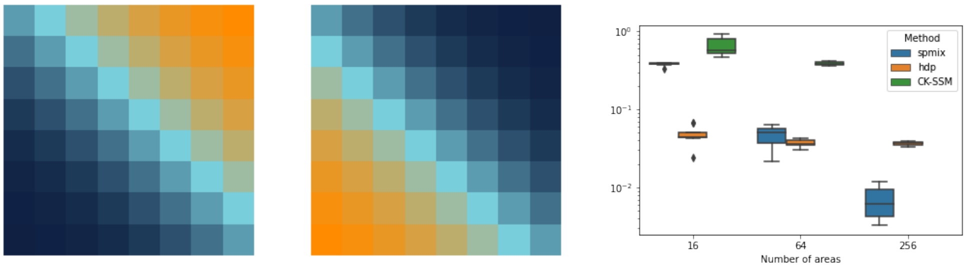
where the weights are chosen as $alr^{-1}(\tilde{\boldsymbol{w}}_i)$ and the transformed weights $\tilde{\boldsymbol{w}}_i$ are given by

$$\tilde{w}_{i1} = 3(x_i - \bar{x}) + 3(y_i - \bar{y}) \quad \tilde{w}_{i2} = -3(x_i - \bar{x}) - 3(y_i - \bar{y}) \tag{7.23}$$

where (x_i, y_i) are the coordinates of the center of area i and $(\bar{x}, \bar{y})$ the coordinates of the grid center. In this way, we introduce strong spatial dependence, induced by (7.23), among the weights of different areas, as we observe in Figures 7.6.3a and 7.6.3b, where we plot the weights of the first two components $\{w_{i1}\}$ and $\{w_{i2}\}$, for the scenario $I = 64$; of course $w_{i3} = 1 - w_{i1} - w_{i2}$.

In our model, we consider areas i and j to be neighbors if they share an edge, setting $g_{ij} = 1$ in (7.9) in this case, and $g_{ij} = 0$ otherwise. For each scenario, we simulated 10 independent datasets, sampling 25 observations per area, and then we compare the posterior estimates of the densities with the true ones via Kullback-Leibler divergence. We compare our model with the HDP-mixture model and CK-SSM, reporting in Figure 7.6.3c the errors, averaged over all areas, for the ten repetitions. Though when $I = 16$ HPD gives much better estimates, our model outperforms the HDP-mixture, when the number of areas is sufficiently large, with consistent results using the Hellinger distance to measure the errors, as shown in Figure 7.D.3 in 7.D. On the other hand, CK-SSM shows the worst posterior estimates for $I = 16, 64$, whereas we do not present the case $I = 256$ for its high computational cost.

Concerning our model (spmix) and the HDP-mixtures, the median execution times, over the 10 datasets, of the code corresponding to spmix was 25.28, 118.14 and 616.41 seconds for $I = 16, 64, 256$, respectively, whereas, for fitting data for the HPD-mixtures



(a) (b) (c)

Figure 7.6.3: Simulation from spatially dependent weights in Sect. 7.6.2: plots of $\{w_{i1}\}$ (a), $\{w_{i2}\}$ (b) when $I = 64$; boxplots (c) of the Kullback-Leibler divergence between true density (7.22) and estimated one under our model (spmix), the HDP-mixture model (hdp) and the CK-SSM

for each simulation, averaged over the areas, for $I = 16, 64, 256$, in logarithmic scale.

was 18.39, 72.59, 207.46 seconds. Based on our implementation, HDP is slightly faster, but our model still exhibits competitive computational times, paired with lower errors when the number of areas is large. Simulations were performed on a machine equipped with a 4x Xeon E5-2640 v4 @2.4GHz processor and 64 GB of RAM. In order to provide a fair comparison, the implementation for our model ran on a single core since the sampler for the HDP is inherently sequential. However, the Gibbs sampler we proposed can be straightforwardly parallelized and this could greatly decrease the runtimes of our model.

7.7 Airbnb Amsterdam

We consider the Airbnb listings dataset for the city of Amsterdam (The Netherlands), publicly available at
http://insideairbnb.com/get-the-data.html. The dataset consists of more than 20,000 listings spread over Amsterdam, grouped by the neighborhood. Our main goal is the prediction of the nightly price of a new listing, with information given by covariates, and taking into account the spatial dependence. As mentioned in the Introduction, (joint) density modeling and estimation, in this case, can give insight to landlords who need to take decision on where their flats should be positioned in the flat rental market. In fact, in this application, uncertainty quantification associated to the prediction, which can be easily derived from the full posterior density estimate, can lead to better informed decisions about the market strategy. We consider two generalizations of model (7.7) to account for covariates as follows. Denote responses as y_{ij} (i.e. the nightly price of accommodation j in neighborhood i) and covariates as $\boldsymbol{x}_{ij} = (x_{ij1}, \ldots, x_{ijd})^T$. In the first model, denoted here $M1$, we assume τ_h in (7.7) such that $\tau_h = (\mu_h + \boldsymbol{\beta}^T \boldsymbol{x}_{ij}, \sigma_h^2)$, $h = 1, \ldots, H$. $M1$ can be understood as a linear regression model with component-specific intercept and variance. We further generalize $M1$ by assuming that all the regression coefficients are component-specific, i.e. $\tau_h = (\mu_h + \boldsymbol{\beta}_h^T \boldsymbol{x}_{ij}, \sigma_h^2)$, $h = 1, \ldots, H$, and denote it by $M2$. While model $M1$ assumes that the effect of the covariates on the pricing is shared across all neighborhoods, and the spatial effect can be represented by the only intercept, model $M2$ assumes that all the covariates have different effect on the pricing depending on the neighborhood.

7.7.1 Data description

We consider as predictors characteristics of the house such as: (i) `accommodates`, the number of guests that can be hosted, (ii) `bathrooms`, the number of bathrooms, (iii) `bedrooms`, the number of bedrooms; together with two indicators of popularity of the listing: (iv) `number_of_reviews` the number of reviews present for that listing, and (v) `review_scores_rating` the average rating of the reviews. Finally, we complete the set of covariates with two binary variables: (vi) `instant_bookable` which equals 1 if the listing can be booked instantly from the user and 0 if, instead, the request must go through an acceptance procedure from the host; (vii) `host_is_superhost` that is 1 if the host is classified as a *superhost* by Airbnb and 0 otherwise. The *superhost* badge can be obtained once a host has a sufficient number of reviews with a rating above a certain threshold. These binary variables were included since the user, while searching for an accommodation, can reduce her/his search only to instant bookable listings and/or only to *superhosts.*

As preprocessing, we used the following steps: we removed the listings for which at least one predictor is missing, as well as listings whose nightly price is below two euros or above one thousand euros; then we transformed `number_of_reviews` by taking the natural logarithm and `review_scores_rating` by the Box-Cox transformation $x_i^{(\lambda)} = (x_i^\lambda - 1)/\lambda$ (Box and Cox, 1964) with $\lambda = 12$, being this value automatically chosen by the Python package `scipy`.

Each listing is assigned to one of the twenty-two Amsterdam neighborhoods provided at the dataset web page, so that $I = 22$. The total number of observations considered for our analysis is $N_1 + \cdots + N_I = 17,201$. Figure 7.7.1(a) shows sample sizes in the log-scale for each neighborhood; of course, city center is the area with the largest number of observations. Furthermore, in Figure 7.7.1, panels (b) and (c), we report sample means and standard deviations of the nightly price in euros in each neighborhood; the plots motivate the modeling of the spatial dependence, as close neighborhoods tend to have similar distributions, at least in terms of mean and standard deviation. Figure 7.7.1 shows that there are two distinct graph connected components, one made only by three areal units; this agrees with official neighborhood maps of the city of Amsterdam. As far as covariates are concerned, Figure 7.D.4 in 7.D shows empirical correlations among the predictors and between predictors and the response. Figure 7.D.5 in 7.D displays scatterplots of the response price versus numerical predictors and boxplots for categorical predictors. We note that only `accommodates`, `bathrooms`, `bedrooms` exhibit a significant linear correlation with the price, which is confirmed by the scatter plots, while sample linear correlation between `accommodates` and `bathrooms` is 0.362, 0.730 between `accommodates` and `bedrooms`, 0.430 between `bathrooms` and `bedrooms`. However, when computing the variance inflation factor, we found 2.197, 1.243 and 2.334, respectively, values that suggest very mild multicollinearity. On the other hand, there is no evident empirical effect of `instant_bookable` and `host_is_superhost` on the nightly price as Figure 7.D.5 in 7.D shows. In the next subsection, we consider both models $M1$ and $M2$ for the dataset, including all the covariates above described, i.e. $d = 7$. We standardized all numerical predictors, subtracting the sample mean and dividing by the sample standard deviation of each predictor; we also centered the response on the overall sample mean.

7.7.2 Posterior inference

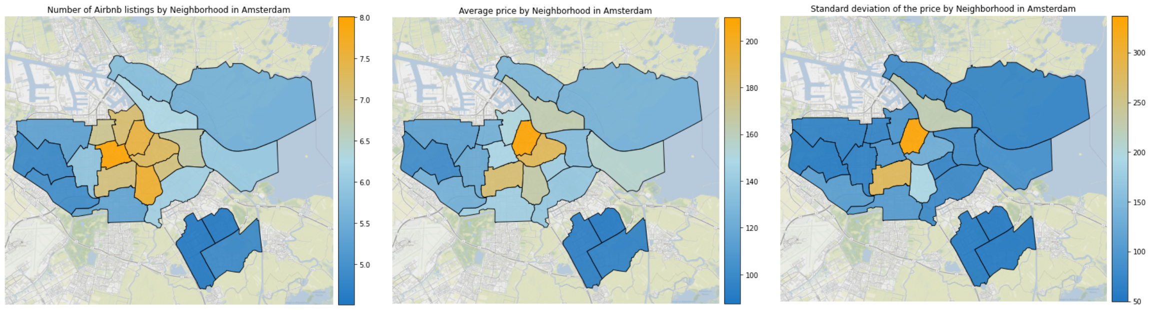
We complete the prior for model $M1$ assuming

$$(\mu_h, \sigma_h^2) \overset{\text{iid}}{\sim} \mathcal{N}(\mu_h \,|\, 0, 2\sigma_h^2) \times IG(\sigma_h^2 \,|\, 2, 2), \quad h = 1, \dots, H$$

and $\boldsymbol{\beta} \sim \mathcal{N}_d(\mathbf{0}, \sigma_\beta^2 \mathbb{I}_d)$. For the prior of model $M2$ we assume

$$((\mu_h, \boldsymbol{\beta}_h), \sigma_h^2) \overset{\text{iid}}{\sim} \mathcal{N}((\mu_h, \boldsymbol{\beta}_h) \,|\, \mathbf{0}, 10\mathbb{I}_{d+1}) \times IG(\sigma_h^2 \,|\, 2, 2), \quad h = 1, \dots, H.$$

We need to change the Gibbs sampler in Section 7.5, adding two further steps to sample from the full conditional of $\boldsymbol{\beta}$ for model $M1$ or from the full conditional of (μ_h, β_h) for


Number of Airbnb listings by Neighborhood in Amsterdam
8.0
7.5
7.0
6.5
6.0
5.5
5.0
Average price by Neighborhood in Amsterdam
200
180
160
140
120
100
Standard deviation of the price by Neighborhood in Amsterdam
300
250
200
150
100
50

(a) No. of listings in the log-scale

(b) Sample mean

(c) Sample standard deviation

Figure 7.7.1: Number of listings (in the log-scale), sample means and standard deviations of the nightly price in euros for each neighborhood in the Airbnb dataset, after preprocessing.

model $M2$. Both steps are standard updates in Bayesian linear regression models; see 7.C for further details.

Posterior inference is robust to the choice of all the hyper-parameters in the prior distribution, but for the number H of components in the mixture, that is a key parameter for mixture models. For this reason, we compare $M1$ and $M2$ via predictive goodness-of-fit indexes such as the log-pseudo marginal likelihood (LPML, Geisser and Eddy, 1979) and the widely applicable information criterion (WAIC, Watanabe, 2013), when H varies in $\{5, 10, 15\}$. Better predictive performances are associated to higher LPML and lower WAIC. In this comparison, we also consider a generalization of the CK-SSM model in Jo et al. (2017) along the lines of model $M1$. Table 7.7.1 shows that the best model is $M1$, across all values of H and that CK-SSM does a worse job than $M1$ and $M2$. Given its superior predictive performance, in the following we consider only $M1$.

In particular, $M1$ with $H = 15$ gives slightly better values of LPML and WAIC, but the difference across all values of H seems negligible so that, to fix H, we consider also the predictive mean squared error computed through a 10-fold cross-validation. The cross-validation is stratified according to the areas, so that, each time, approximately 10% of the data is missing from each neighborhood. More in detail, each time we select 90% of the dataset as 'training set' (to simulate from the relative posterior) and compute the mean of the predictive distribution corresponding to data in the 'testing set'. Observe that the same datapoints are shared across all values of H, both for training and for testing. Then we compute the predictive mean squared error (pMSE) on the testing set, i.e. $\sum_{i=1}^{m}(y_i - \hat{y}_i)^2/m$, where m is the size of the testing set and $\hat{y}_i$ is the mean of the posterior predictive density of the response corresponding to covariate $\boldsymbol{x}_i$. The average cross validation error ± one standard deviation is equal to 5468 ± 952, 5474 ± 850 and 5477 ± 956 for $H = 5, 10, 15$ respectively.

We have also considered the case $H = 1$, i.e., when all $M1$, $M2$ and CK-SSM models are equivalent to a standard Gaussian linear regression In this case, the predictive performance is much worse (LMPL and WAIC are approximately equal to -1.3×10^5 and 2.6×10^5 respectively), hence showing that a richer model with explicit modeling of the spatial dependence structure is needed to obtain better predictive performances.

Lastly, removing covariates `bathrooms` and `bedrooms`, correlated with `accommodates`, resulted in slightly worse predictive performance for all models tested; for instance, $M1$ showed a decrease in LMPL of 2.5%, while for $M2$ the decrease was around 1% across all values of H. Summing up, for the reasons above, including parsimony of the model, in the rest of the section, we consider only model $M1$ when $H = 5$.

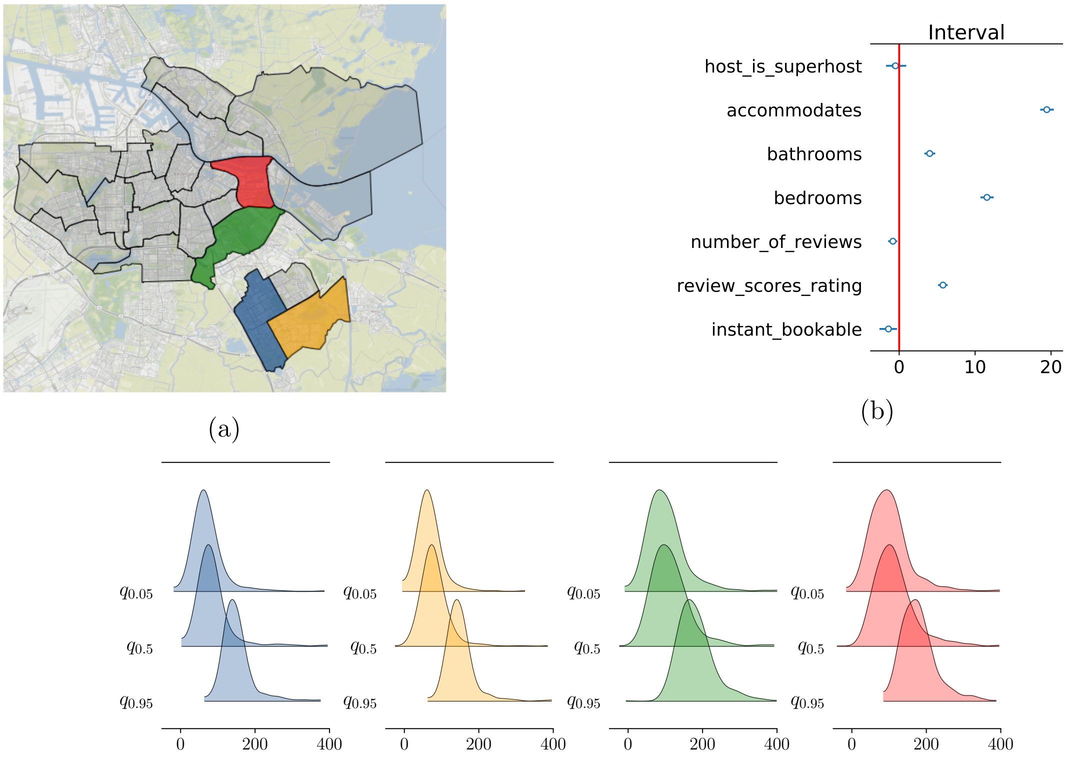
Further comparisons are discussed in 7.E. We first analyze this dataset by modeling connected components separately. Then we compare also with the fit given by geograph-

	$M1$			$M2$			CK-SSM		
H	5	10	15	5	10	15	5	10	15
LPML	-92619	-92444	-92441	-97998	-97836	-97828	-98751	-98752	-98755
WAIC	185238	184888	184882	195996	195672	195656	197502	197504	197505

Table 7.7.1: LPML and WAIC for various choices of H under $M1$, $M2$ and CK-SSM.

ically weighted regression. In both cases, the predictive performances are worse than the ones obtained with our model.

Figure 7.7.2(b) reports 95% posterior credibility intervals for the regression parameters. All the covariates, except for `host_is_superhost`, seem to be significant, if we assume hard shrinkage as a criterion for significance, i.e. the marginal credibility interval does not include 0. It is interesting to observe how the coefficients associated to `number_of_reviews` and `instant_bookable` are negative. This might indicate that hosts that receive many reviews and many reservations tend to lower their prices in order be more attractive. On the other hand, as one would expect, all the other coefficients are positive, the one furthest right being associated to `accommodates`, i.e. the number of guests that can be hosted. In Figure 7.7.2(c) we show the density estimates in the four neighborhoods highlighted in



(a)

(b)

(c)

Figure 7.7.2: (a): Map of the city of Amsterdam with neighborhoods *Bijlmer-Centrum* in blue, *Gaasperdam - Driemond* in orange, *Oostelijk Havengebied - Indische Buurt* in green and *Watergraafsmeer* in red. (b): 95% credibility intervals of the marginal posterior of the regression parameter $\boldsymbol{\beta}$. (c): Predictive densities for different neighborhoods, the colors match the ones of the map. In each plot, three lines represent three different values of the covariates `accommodates`, `number_of_bedrooms`, `number_of_bathrooms`), equal to their 5%, 50% and 95% sample quantiles, while the other numerical covariates are fixed to the empirical median.

	Bijlmer Centrum			Gaasperdam Driemond			Oostelijk Havengebied Indische Buurt			Watergraafsmeer		
	$q_{0.05}$	$q_{0.50}$	$q_{0.95}$	$q_{0.05}$	$q_{0.50}$	$q_{0.95}$	$q_{0.05}$	$q_{0.50}$	$q_{0.95}$	$q_{0.05}$	$q_{0.50}$	$q_{0.95}$
mean	71.5	84.23	149.5	69.38	82.18	147.8	99.49	112.1	176.5	100.0	112.6	177.0
median	66.53	79.35	145.9	64.92	77.75	145.1	92.18	105.0	172.33	92.98	105.8	173.1
sd	40.73	40.38	39.93	37.59	37.30	36.86	52.85	52.67	53.30	52.89	52.711	53.37
P_{200}	0.02	0.02	0.06	0.01	0.01	0.05	0.05	0.06	0.26	0.05	0.06	0.27

Table 7.7.2: Summary statistics (mean, median, standard deviation and probability of exceeding 200 euros) of the posterior predictive distributions for the same neighborhoods and covariate choices as in Figure 7.7.2.

Figure 7.7.2(a). Each plot shows three density estimates, corresponding to different values of the covariates. In this case, the covariates were set to the empirical median except for `accommodates`, `number_of_bedrooms`, `number_of_bathrooms`. Since the marginal sample correlation between these three covariates is not negligible as mentioned in Section 7.7.1, we have fixed all their values simultaneously equal to 5%, 50% and 95% empirical quantiles, respectively. For instance, in each panel of Figure 7.7.2(c), the top lines correspond to density estimates for a vector of covariates in which `accommodates`, `number_of_bedrooms`, `number_of_bathrooms` are fixed to their 5% sample quantile, respectively. It is clear from Figure 7.7.2(c) that the predictive densities in blue (first panel from the left) and in yellow are similar, as well as the lines in green and in red. However there are evident differences when comparing for instance the yellow densities (second panel from the left) with the green ones (third panel from the left); indeed the green densities give substantial mass to the right tail, especially to values greater than 200 euros, while the yellow densities do not. This behavior agrees with the marginal posterior of ρ, that is strongly concentrated near 1 ($\mathbb{E}(\rho|data) = 0.993$): in fact, the blue and yellow neighborhoods, as well as the red and green ones, are connected in the graph. However, blue and yellow predictive densities are different from the green and red estimates, since the neighborhoods belong to different connected components in the graph. As expected, in all the neighborhoods the listings price increases as the `accommodates`, `number_of_bedrooms`, `number_of_bathrooms` increase as well.

To improve understanding of the posterior predictive densities evaluated on a grid of points as in Figure 7.7.2(c), which are, by definition, the posterior means of the likelihood function, we focus on different summary statistics of these distributions. In particular, for any of the selected areas as in Figure 7.7.2, we focus on the posterior predictive mean, median, standard deviation of the nightly price and on the posterior predictive probability $P(y^\star > 200 \,|\, \boldsymbol{x}^\star, i)$ that the nightly price exceeds 200 euros, conditioning on the covariates selected above. Numerical values for the summary statistics are reported in Table 7.7.2, where for each of the four selected neighborhoods we consider three different values of the covariates $\boldsymbol{x}^\star$, denoted by $q_{0.05}, q_{0.5}$ and $q_{0.95}$ (see the discussion above and the caption of Figure 7.7.2 for their definition). It is clear that for each of the neighborhoods, the mean and median of the price increase as `accommodates`, `number_of_bedrooms`, `number_of_bathrooms` increase as well. Interestingly, $P(y^\star > 200 \,|\, \boldsymbol{x}^\star, i)$ increases only slightly when the covariates go from $q_{0.05}$ to $q_{0.50}$ but increases significantly when covariates are $q_{0.95}$. As expected from Figure 7.7.2(c), for each value of covariates, summary statistics in Bijlmer Centrum and Gaasperdam Driemond assume close values. A similar comment applies to summary statistics in Oostelijk Havengebied Indische Buurt and Watergraafsmeer. Instead, larger differences are observed in Table 7.7.2 when comparing Bijlmer Centrum and Watergraafsmeer (corresponding to the blue and red densities in Figure 7.7.2(c)).

To conclude, we believe that a new lessor could benefit from our analysis because Airbnb makes available only a handful of covariates which might not be suited to fully characterize the ‘right’ nightly price for a listing. For instance, we expect that the presence of a balcony or garden might lead to higher prices. Hence, when deciding the price for a listing, the lessor could look at the predictive distribution from our model given the covariates and neighborhood of their house, and choose to place the listing in the right or left tail of the predictive distribution considering additional information not included in our model.

7.8 Discussion

In this work, we have considered the problem of the joint estimation of spatially dependent densities in the context of repeated areal measurements. We have presented a finite mixture model to represent the density in each area; assuming that all the mixtures share the same set of atoms, the spatial dependency has been introduced through a novel joint distribution for I vectors in the simplex as a prior for the mixture weights. This distribution, that we termed logisticMCAR, was built as a logistic transformation of a specification of the multivariate CAR model. When compared to alternatives proposed in the literature, the logisticMCAR distribution showed to have a higher degree of interpretability, as we were able to derive the analytic expression for the expected values of ratios of components and their covariances, via the Aitchison geometry. Moreover, we also showed as the logisticMCAR can be used to accurately model sparse mixtures.

Posterior simulation has been carried out by means of a Gibbs sampler scheme. In particular the update of the mixture weights was performed by introducing a data augmentation scheme based on the Pólya-Gamma identity, which avoids the tedious tuning of the proposal distribution.

In the simulation studies and the real application included in this chapter, our model has shown to be able to represent a wide range of different behaviors. In particular, we argue that when different connected graph components are present, and heterogeneous behavior is observed across these components, our model should be preferred as it does not force the densities in different graph component to behave too similarly. Moreover, as in the case of the Airbnb Amsterdam application, our model can be easily extended to include additional covariate information. Although not our target here, a sub-product of the approach is the prior induced on the partition of the subjects in the sample, which in this case, has a spatial connotation; relations with spatial product partition models (Page and Quintana, 2016) could be further investigated.

Another point that we did not address here, and will be focus of future study, is an extension to models where the *graph* G is not fixed, and should be learned by the data (and the prior). In particular, we aim at considering boundary detection problems, i.e. when the proximity matrix G is unknown, but its elements depend on dissimilarity metrics available for each pair of neighboring areas. This is an extremely interesting problem, widely studied in the context of one single response per area; see, for instance Lu et al. (2007) and Lee and Mitchell (2012). However preliminary investigation showed how the non-identifiability of overfitted mixtures might produce erroneous results. Possible extensions of our model to account for boundary detection might then include either a prior on the number of components or a repulsive prior distribution on the atoms, or both, to reduce the impact of non-identifiability.

Appendix

7.A Proofs

Proof of Proposition 7.1

The proof proceeds along the lines of Section 2 in Mardia (1988) where a similar MCAR model is proposed. Briefly, we need only to show that the conditional distributions of $\tilde{\boldsymbol{w}}_i \,|\, \tilde{\boldsymbol{w}}_{-i}$ derived from the joint distribution (7.5) coincide with the one given by (7.4), since by the Hammersley–Clifford theorem the set of full conditionals will uniquely identify the joint distribution in our setting; see also Theorem 2.1 in Mardia (1988). We start by assuming that only one connected component is present in G and that, without loss of generality, $\tilde{\boldsymbol{m}}_i = \tilde{\boldsymbol{m}}_j = \boldsymbol{0}$ for each $i, j = 1, \ldots, I$. The more general case, i.e., when $\tilde{\boldsymbol{m}}_i$ is not zero (but $\tilde{\boldsymbol{m}}_i = \tilde{\boldsymbol{m}}_j$ if i and j belong to the same connected component), follows from the same calculations below replacing $\tilde{\boldsymbol{w}}_i$ with $\tilde{\boldsymbol{w}}_i - \tilde{\boldsymbol{m}}_i$. For ease of notation we focus here on the full conditional of $\tilde{\boldsymbol{w}}_1$ but the other distributions can be derived in a similar manner.

From (7.5), the joint density of $\tilde{\boldsymbol{w}}$ is proportional to:

$$\exp\left(-\frac{1}{2}\tilde{\boldsymbol{w}}^T\left((F-\rho G)\otimes\Sigma^{-1}\right)\tilde{\boldsymbol{w}}\right).$$

We rewrite the quadratic form above to highlight only what depends on $\tilde{\boldsymbol{w}}_1$:

$$(\tilde{\boldsymbol{w}}_1 \cdots \tilde{\boldsymbol{w}}_I)\left(\begin{array}{c|c|c} (F-\rho G)_{11}\Sigma^{-1} & \cdots\cdots & (F-\rho G)_{1I}\Sigma^{-1} \\ \hline \vdots & & \vdots \\ \hline (F-\rho G)_{I1}\Sigma^{-1} & \cdots\cdots & (F-\rho G)_{II}\Sigma^{-1} \end{array}\right)\begin{pmatrix}\tilde{\boldsymbol{w}}_1 \\ \vdots \\ \tilde{\boldsymbol{w}}_I\end{pmatrix}$$

where $(\tilde{\boldsymbol{w}}_1 \cdots \tilde{\boldsymbol{w}}_I)$ corresponds to the vectorization of the $\tilde{w}_i$'s. Thanks to this block structure we obtain

$$\begin{aligned}
&\exp\left(-\frac{1}{2}\tilde{\boldsymbol{w}}^T\left((F-\rho G)\otimes\Sigma^{-1}\right)\tilde{\boldsymbol{w}}\right) \\
&\qquad = \exp\Big\{-\frac{1}{2}\Big(\tilde{\boldsymbol{w}}_1^T(F-\rho G)_{11}\Sigma^{-1}\tilde{\boldsymbol{w}}_1 + \\
&\qquad\qquad + 2\tilde{\boldsymbol{w}}_1^T[(F-\rho G)_{12}\cdots(F-\rho G)_{1I}])\otimes\Sigma^{-1}[\tilde{\boldsymbol{w}}_2\cdots\tilde{\boldsymbol{w}}_I]^T \\
&\qquad\qquad + [\tilde{\boldsymbol{w}}_2\cdots\tilde{\boldsymbol{w}}_I]K[\tilde{\boldsymbol{w}}_2\cdots\tilde{\boldsymbol{w}}_I]\Big)\Big\}
\end{aligned} \tag{7.24}$$

for some matrix K, which is irrelevant for our purposes since it does not interact with $\tilde{\boldsymbol{w}}_1$ so that the term depending on K can be discarded. Hence, we are able to identify in (7.24) the quadratic form of a Gaussian distribution and conclude that the full conditional of $\tilde{\boldsymbol{w}}_1$ is Gaussian. In particular, the first addend in the right hand-side of (7.24) is the only term that is quadratic in $\tilde{\boldsymbol{w}}_1$, so that it must be equal to $\tilde{\boldsymbol{w}}_1 S^{-1}\tilde{\boldsymbol{w}}$ where S is the covariance matrix of the full conditional of $\tilde{\boldsymbol{w}}_1$. Similarly, the second addend can be expressed as $-2\tilde{\boldsymbol{w}}_1 S^{-1}\boldsymbol{\mu}$ where $\boldsymbol{\mu}$ is the mean of the Gaussian distribution. Some linear algebra yields

$$\begin{aligned}\mathrm{Var}[\tilde{\boldsymbol{w}}_1 \mid \tilde{\boldsymbol{w}}_{-1}] &= S = \left((F-\rho G)_{11}\Sigma^{-1}\right)^{-1} = \frac{\Sigma}{\rho\sum_{j=1}^{I} g_{ij} + 1 - \rho} \\ \mathbb{E}[\tilde{\boldsymbol{w}}_1 \mid \tilde{\boldsymbol{w}}_{-1}] &= \boldsymbol{\mu} = \frac{\rho\sum_{j=1}^{I} g_{ij}\tilde{\boldsymbol{w}}_j}{\rho\sum_{j=1}^{I} g_{ij} + 1 - \rho}.\end{aligned}$$

When there are at least two connected components in the graph, we note that full independence holds across each pair of $\tilde{\boldsymbol{w}}_i$'s as long as the two vectors belong to different connected components. Hence, the same argument above can be carried out on each single connected component. The joint density still takes the same form because the matrix $(F-\rho G)\otimes\Sigma^{-1}$ is block diagonal in this case with each block corresponding to one of the connected components. □

Proof of Proposition 7.2

From equation (7.4) we have that

$$\begin{aligned}\mathbb{E}\left[\tilde{\boldsymbol{w}}_i \mid \tilde{\boldsymbol{w}}_{-i}\right] &= \frac{\rho\sum_{j\in U_i}\tilde{\boldsymbol{w}}_j + (1-\rho)\tilde{\boldsymbol{m}}_i}{\rho|U_i| + 1 - \rho} = \frac{\rho\sum_{j\in U_i} alr(\boldsymbol{w}_j) + (1-\rho)alr(\boldsymbol{m}_i)}{\rho|U_i| + 1 - \rho} \\ &= \frac{1}{\rho|U_i|+1-\rho}\left(\log\frac{\prod_{j\in U_i} w_{j1}^{\rho} m_{i1}^{1-\rho}}{\prod_{j\in U_i} w_{jH}^{\rho} m_{iH}^{1-\rho}}, \dots, \log\frac{\prod_{j\in U_i} w_{jH-1}^{\rho} m_{iH-1}^{1-\rho}}{\prod_{j\in U_i} w_{jH}^{\rho} m_{iH}^{1-\rho}}\right) \\ &= \frac{1}{\rho|U_i|+1-\rho}\left(\sum_{j\in U_i}\log(w_{j1}^{\rho} m_{i1}^{1-\rho}), \dots, \sum_{j\in U_i}\log(w_{jH-1}^{\rho} m_{iH-1}^{1-\rho})\right) \\ &- \sum_{j\in U_i}\log(w_{jH}^{\rho} m_{iH}^{1-\rho})\end{aligned}$$

where the last subtraction is meant elementwise. Hence we have that

$$\begin{aligned}\mathbb{E}\left[\log\frac{w_{il}}{w_{ik}} \mid \boldsymbol{w}_{-i}\right] &= \mathbb{E}\left[\tilde{w}_{il} - \tilde{w}_{ik} \mid \boldsymbol{w}_{-i}\right] \\ &= \frac{1}{\rho\,|U_i| + 1 - \rho}\left(\sum_{j\in U_i}\log(w_{jl}^{\rho} m_{il}^{1-\rho}) - \sum_{j\in U_i}\log(w_{jk}^{\rho} m_{ik}^{1-\rho})\right) \\ &= \log\left(\left(\frac{m_{il}}{m_{ik}}\right)^{1-\rho}\prod_{j\in U_i}\left(\frac{w_{jl}}{w_{jk}}\right)^{\rho}\right)^{\frac{1}{\rho|U_i|+1-\rho}}\end{aligned}$$

which proves the proposition. □

Proof of Proposition 7.3

The (marginal) joint distribution of two different components of $\tilde{\boldsymbol{w}}_i, \tilde{\boldsymbol{w}}_j$, with $i, j = 1, \dots, I$, $i \neq j$ can be easily derived from (7.5):

$$\begin{pmatrix}\tilde{w}_{il}\\ \tilde{w}_{jm}\end{pmatrix} \sim \mathcal{N}_2\left(\boldsymbol{0}, \begin{bmatrix}A_{ii}\Sigma_{ll} & A_{ij}\Sigma_{lm}\\ A_{ji}\Sigma_{ml} & A_{jj}\Sigma_{mm}\end{bmatrix}\right) \qquad l, m = 1, \dots, H-1$$

Hence, we compute the covariance of the log ratios of different components as

$$\begin{aligned}\mathrm{Cov}\left(\log\frac{w_{il}}{w_{im}},\log\frac{w_{jl}}{w_{jm}}\right) &= \mathrm{Cov}\left(\tilde{w}_{il}-\tilde{w}_{im},\ \tilde{w}_{jl}-\tilde{w}_{jm}\right)\\ &= \mathrm{Cov}\left(\tilde{w}_{il},\tilde{w}_{jl}\right)+\mathrm{Cov}\left(\tilde{w}_{il},\tilde{w}_{jm}\right)++\mathrm{Cov}\left(\tilde{w}_{im},\tilde{w}_{jl}\right)+\mathrm{Cov}\left(\tilde{w}_{im},\tilde{w}_{jm}\right)\\ &= A_{ij}\left(\Sigma_{ll}-2\Sigma_{lm}+\Sigma_{mm}\right)\end{aligned}$$

whereas, for the last component,

$$\mathrm{Cov}\left(\log\frac{w_{il}}{w_{iH}},\log\frac{w_{jl}}{w_{jH}}\right)=\mathrm{Cov}\left(\tilde{w}_{il},\ \tilde{w}_{jl}\right)=A_{ij}\Sigma_{ll}$$

which proves the formula in the proposition.

It is possible to rearrange the indices $1,\dots,I$ in order for $(F-\rho G)$ to be a block diagonal matrix, where each block corresponds to a connected graph component according to the neighboring structure; this will not affect the joint law. By the properties of strictly diagonally dominated matrices, the same pattern of blocks is preserved in the inverse matrix A. Hence $A_{ij}=0$ if i and j belong to two non-connected graph components, proving the proposition thanks to equivalence between uncorrelation and independence of Gaussian random variables. □

7.B MC simulations from the logisticMCAR distribution

In Section 7.3 we have pointed out that the theoretical analysis of the logisticMCAR distribution is limited by its analytic intractability. Here we compute covariances between different components of the vectors of weights and Euclidean distances between the vectors themselves through Monte Carlo simulation. Specifically, we simulate from (7.5) and then obtain draws from the logisticMCAR distribution through the transformation alr^{-1}.

In particular, we fix $I=5$, $H=3$, $\tilde{\boldsymbol{m}}_i=0$ for all i and the covariance matrix Σ

$$\Sigma=\begin{bmatrix}1 & \Sigma_{12}\\ \Sigma_{12} & 1\end{bmatrix}$$

where Σ_{12} denotes the covariance, but also the correlation since $\Sigma_{11}=\Sigma_{22}=1$, between $\tilde{w}_{i1}$ and $\tilde{w}_{i2}$. We fix the proximity matrix G such that $g_{12}=g_{13}=g_{23}=1$ and $g_{45}=1$. This corresponds assuming that areal units/nodes 1, 2 and 3 are connected to each other, and 4 and 5 are connected to each other, though separated from the others.

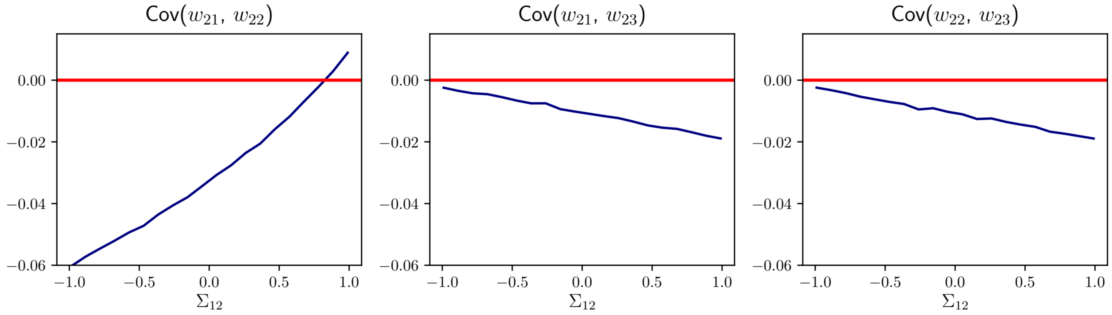


Figure 7.B.1: Pairwise covariance values of the components of $\boldsymbol{w}_2=(w_{21},w_{22},w_{23})$ as a function of the correlation parameter Σ_{12}. The horizontal red line indicates the value 0.

Figure 7.B.1 shows the covariance between the three components of $\boldsymbol{w}_2 = (w_{21}, w_{22}, w_{23})$ as a function of the correlation parameter Σ_{12} in the matrix Σ in (7.5), having simulated $N = 10,000$ MC draws. Note that, unlike the finite-dimensional Dirichlet distribution, the logistic-normal distribution may have positive covariance among the components.

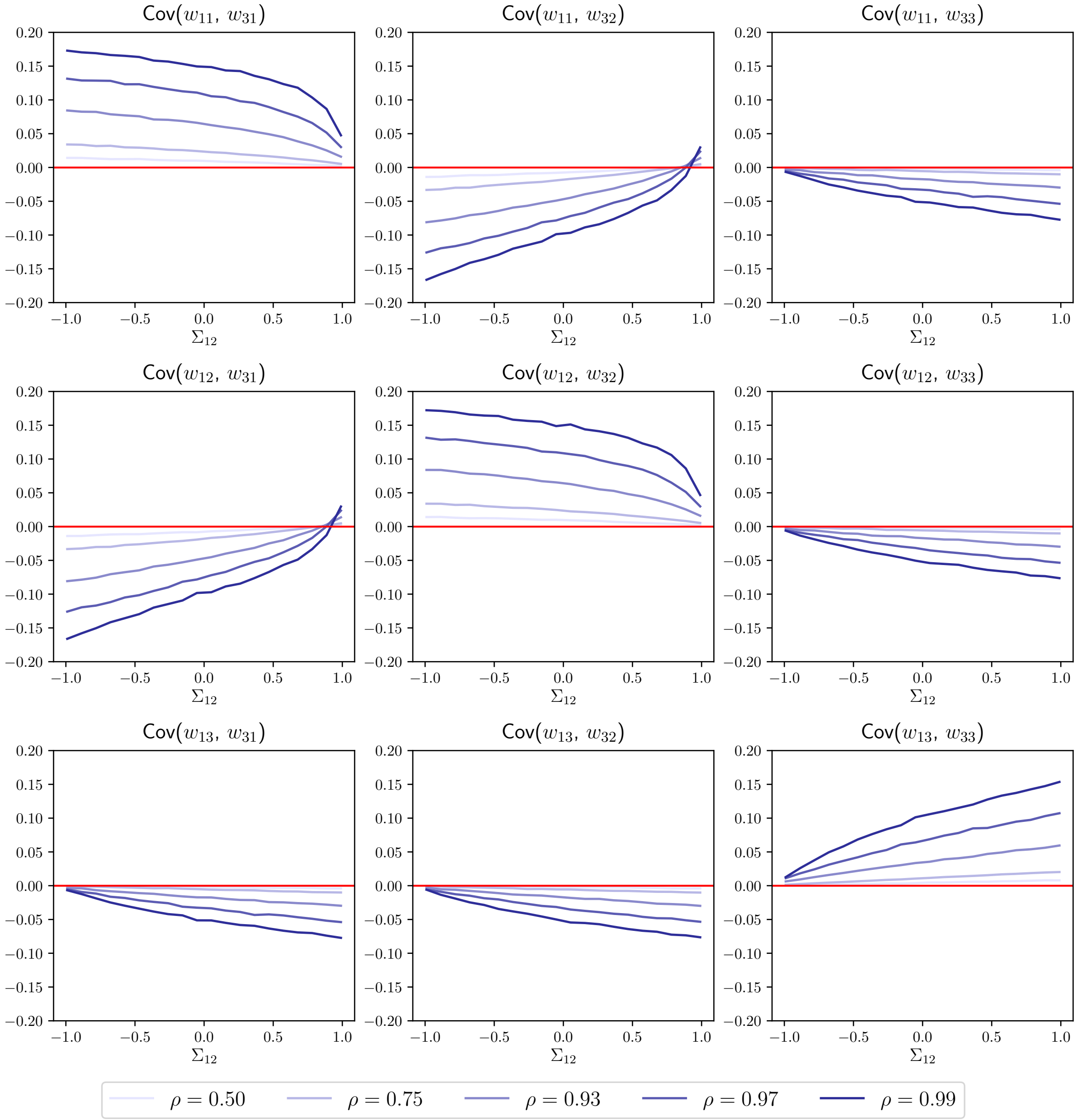


Figure 7.B.2: Pairwise covariance values between components of $\boldsymbol{w}_1$ and $\boldsymbol{w}_3$, as a function of the correlation parameter Σ_{12}, for different values of ρ, when $(\boldsymbol{w}_1, \dots, \boldsymbol{w}_5)$ has the logisticMCAR distribution. The horizontal red line indicates the value 0.

Figure 7.B.2 instead shows the covariance between all the possible pairs (w_{1j}, w_{3m}) for $j, m = 1, 2, 3$, for different values of the parameter ρ. The covariances between corresponding entries, i.e. $(w_{1j}, w_{2j}\ j = 1, 2, 3)$ is always positive, as expected since the spatial correlation parameter ρ is always fixed to a positive value. The marginal prior for $\boldsymbol{w}_1, \boldsymbol{w}_3$ is exchangeable, since nodes 1 and 3 belong to the same connected component in G. This explains the symmetries in Figure 7.B.2.

In order to measure the association induced by our logisticMCAR prior, we simulate the distances (Euclidean) of two vectors drawn from the joint distribution. In particular, we

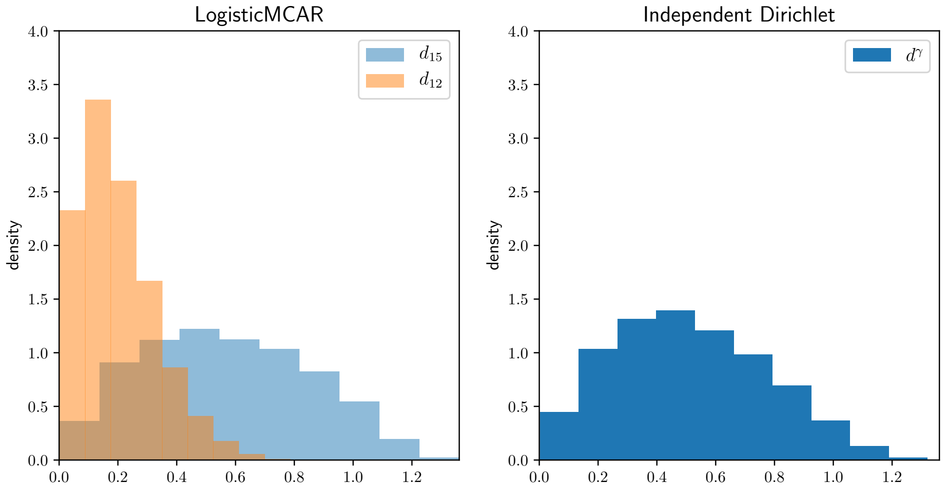


Figure 7.B.3: Histogram of MC draws from the marginal distributions of d_{12} (orange) and d_{15} (light blue) on the left and from the marginal distribution of d^{γ} on the right.

	min	$q_{0.25}$	$q_{0.5}$	$q_{0.75}$	max
d_{12}	4×10^{-4}	0.10	0.18	0.27	0.86
d_{15}	0.01	0.33	0.55	0.77	1.36
d_{γ}	0.007	0.31	0.52	0.71	1.36

Table 7.B.1: Summary statistics of the marginal distributions of the distances $d_{12}, d_{15}, d_{\gamma}$, estimated from the MC samples; q_{α} denotes the α-quantile.

simulated $N = 10,000$ draws from the full joint logisticMCAR distribution of $(\boldsymbol{w}_1, \dots, \boldsymbol{w}_5)$ with parameters as above, fixing $\Sigma_{12} = 0.5$, and computed the Euclidean distances $d_{12} = ||\boldsymbol{w}_1 - \boldsymbol{w}_2||$ and $d_{15} = ||\boldsymbol{w}_1 - \boldsymbol{w}_5||$. As $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ belong to the same connected graph component while $\boldsymbol{w}_5$ belongs to another component, we expect $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ to be more similar than $\boldsymbol{w}_1$ and $\boldsymbol{w}_5$ belonging to separate components. Hence the distance d_{12} should be smaller than d_{15}. Moreover, for comparison, we also simulated $N = 10,000$ draws from the joint distribution of two independent finite-dimensional Dirichlet random variables, i.e.

$$(\gamma_1, \gamma_2)_i \overset{\text{iid}}{\sim} \text{Dir}(\mathbf{1}) \times \text{Dir}(\mathbf{1}) \quad i = 1, \dots N$$

and computed their Euclidean distance as well, that we denote by d^{γ}. Figure 7.B.3 reports the histograms of the marginal distributions of d_{12}, d_{15} on the left and d^{γ} on the right. It is clear that d_{12} is substantially smaller than d_{15}, as expected. Moreover, by comparing d_{15} and d^{γ}, we see that their marginal distributions are very similar. See also the summary statistics of these marginal distributions in Table 7.B.1.

For more insight, we report a subsample of size $N = 20$ of the MC simulated values from the marginal distributions of $(\boldsymbol{w}_1, \boldsymbol{w}_2)$ and (γ_1, γ_2), plotted on the two dimensional projection of the simplex S^3 in Figure 7.B.4. Each pair is denoted by two points inside the triangle and a line connecting them. It is clear that simulated values from $\mathcal{L}(\boldsymbol{w}_1, \boldsymbol{w}_2)_i$ are much closer each other than those from $\mathcal{L}(\gamma_1, \gamma_2)$.

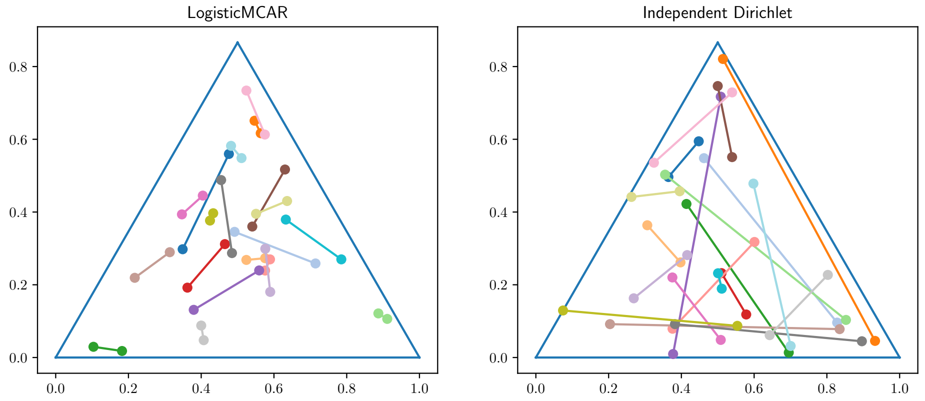


Figure 7.B.4: Plots of $N = 20$ MC draws from the logisticMCAR distribution (left) and of $N = 20$ MC draws from $\mathrm{Dir}(\mathbf{1}) \times \mathrm{Dir}(\mathbf{1})$ (right). Each draws is represented with two dots (the values of the two random vectors) together with a colored line connecting them for visual purposes. Different colors correspond to independent draws.

7.C The Gibbs sampler

Proof of Equations (7.20) **-** (7.21)

We start by writing the full conditional distribution (7.18) as follows:

$$\begin{aligned}\mathcal{L}(\tilde{w}_{ih}\,|\,\tilde{\boldsymbol{w}}_{-i}, \tilde{\boldsymbol{w}}_{i,-h}, \boldsymbol{s}_i, \rho, \Sigma) &\propto \mathcal{N}(\tilde{w}_{ih}\,|\,\mu^*_{ih}, \Sigma^*_{ih}) \left(\frac{\mathrm{e}^{\eta_{ih}}}{1+\mathrm{e}^{\eta_{ih}}}\right)^{N_{ih}} \left(\frac{1}{1+\mathrm{e}^{\eta_{ih}}}\right)^{N_i - N_{ih}} \\ &\propto \mathcal{N}(\tilde{w}_{ih}\,|\,\mu^*_{ih}, \Sigma^*_{ih}) \frac{(\mathrm{e}^{\eta_{ih}})^{N_{ih}}}{(1+\mathrm{e}^{\eta_{ih}})^{N_i}} \\ &\propto \mathcal{N}(\tilde{w}_{ih}\,|\,\mu^*_{ih}, \Sigma^*_{ih})\, \mathrm{e}^{(N_{ih}-N_i/2)\eta_{ih}} \int_0^\infty \mathrm{e}^{-\omega_{ih}\eta^2_{ih}/2} p(\omega_{ih}) d\omega_{ih}\end{aligned}$$

where $\omega_{ih} \sim PG(N_i, 0)$. We now include the latent variable ω_{ih} and derive the conditional distribution of $\tilde{w}_{ih}$, conditioning also to ω_{ih}. We have

$$\begin{aligned}\mathcal{L}(\tilde{w}_{ih}\,|\,\tilde{\boldsymbol{w}}_{-i}, \tilde{\boldsymbol{w}}_{i,-h}, \boldsymbol{s}_i, \rho, \Sigma, \omega_{ih}) &\propto \mathcal{N}(\tilde{w}_{ih}\,|\,\mu^*_{ih}, \Sigma^*_{ih})\, \mathrm{e}^{(N_{ih}-N_i/2)\eta_{ih}} \mathrm{e}^{-\omega_{ih}\eta^2_{ih}/2} \\ &\propto \mathrm{e}^{-\frac{E}{2}}\end{aligned}$$

where

$$\begin{aligned}E &= \frac{(\tilde{w}_{ih} - \mu^*_{ih})^2}{\Sigma^*_{ih}} - (2N_{ih} - N_i)(\tilde{w}_{ih} - C_{ih}) + \omega_{ih}(\tilde{w}_{ih} - C_{ih})^2 \\ &\propto \tilde{w}^2_{ih}\left(\frac{1}{\Sigma^*_{ih}} + \omega_{ih}\right) - 2\tilde{w}_{ih}\left(\frac{\mu^*_{ih}}{\Sigma^*_{ih}} + N_{ih} - N_i/2 + \omega_{ih}C_{ih}\right) \\ &\propto \left(\frac{1}{\Sigma^*_{ih}} + \omega_{ih}\right)\left(\tilde{w}^2_{ih} - 2\tilde{w}_{ih}\left(\frac{\mu^*_{ih}}{\Sigma^*_{ih}} + N_{ih} - N_i/2 + \omega_{ih}C_{ih}\right)\left(\frac{1}{\Sigma^*_{ih}} + \omega_{ih}\right)^{-1}\right)\end{aligned}$$

Thus

$$\mathcal{L}(\tilde{w}_{ih}\,|\,\tilde{\boldsymbol{w}}_{-i}, \tilde{\boldsymbol{w}}_{i,-h}, \boldsymbol{s}_i, \rho, \Sigma, \omega_{ih}) \sim \mathcal{N}(\hat{\mu}_{ih}, \hat{\Sigma}_{ih})$$

where

$$\hat{\mu}_{ih} = \left(\frac{\mu^*_{ih}}{\Sigma^*_h} + N_{ih} - N_i/2 + \omega_{ih} C_{ih}\right)\left(\frac{1}{\Sigma^*_h} + \omega_{ih}\right)^{-1} \quad \hat{\Sigma}_{ih} = \left(\frac{1}{\Sigma^*_h} + \omega_{ih}\right)^{-1}$$

For the full conditional of ω_{ih} instead, it is sufficient to apply Theorem 1 in Polson et al. (2013) with $\psi = \eta_{ih}$ to obtain that the law of ω_{ih}, conditional to $\tilde{\boldsymbol{w}}_i$ is a Pólya-Gamma distribution, i.e. the density of ω_{ih} can be expressed as in Equation (7.19), with parameters $b = N_i$, $c = \tilde{w}_{ih} - \log \sum_{k \neq h} \exp(\tilde{w}_{ik})$.

□

Detailed description of the Gibbs sampler

The state of the MCMC sampler is made of $\boldsymbol{\tau} = (\tau_1, \dots, \tau_H)$, $(\tilde{\boldsymbol{w}}_1, \dots, \tilde{\boldsymbol{w}}_I)$, where $\tilde{\boldsymbol{w}}_i = \text{alr}(\boldsymbol{w}_i)$, $\{s_{ij}\}_{ij}$ and $\tilde{\boldsymbol{m}}_{C_1}, \dots \tilde{\boldsymbol{m}}_{C_k}$. The Gibbs sampler is obtained repeatedly sampling from the following conditional distributions:

- For any $i = 1, \dots, I$ and $j = 1, \dots, N_i$, independently update the cluster allocation variables from

$$p(s_{ij} = h \,|\, rest) \propto \text{alr}^{-1}(\tilde{w}_{ih})\, k(y_{ij} \,|\, \tau_h) \quad h = 1, \dots, H$$

- Independently update the atoms of the mixture from

$$\mathcal{L}(\tau_h \,|\, rest) \propto P_0(\tau_h) \prod_{ij : s_{ij} = h} k(y_{ij} \,|\, \tau_h) \quad h = 1, \dots, H$$

- Sample Σ from

$$\mathcal{L}(\Sigma \,|\, rest) \propto \mathcal{L}(\tilde{\boldsymbol{w}} \,|\, rest)\mathcal{L}(\Sigma)$$

  We show that the full conditional of Σ is still an inverse-Wishart distribution. To see this, write the right hand side as follows

$$\begin{aligned}\mathcal{L}(\Sigma \,|\, rest) \propto & |(F - \rho G)^{-1} \otimes \Sigma|^{-1/2} \exp\left(-\frac{1}{2}(\tilde{\boldsymbol{w}} - \tilde{\boldsymbol{m}})^T \left((F - \rho G) \otimes \Sigma^{-1}\right)(\tilde{\boldsymbol{w}} - \tilde{\boldsymbol{m}})\right) \\ & \times |\Sigma|^{-(\nu + (H-1) + 1)/2} \exp\left(-\frac{1}{2} tr(V \Sigma^{-1})\right)\end{aligned}$$

  Now $|(F - \rho G)^{-1} \otimes \Sigma| = |(F - \rho G)^{-1}|^{H-1} \times |\Sigma|^I$, so that the degrees of freedom in the full conditional are $\nu_p = \nu + I$. Working on the exponent, write the quadratic form involving the Kronecker product as follows

$$(\tilde{\boldsymbol{w}} - \tilde{\boldsymbol{m}})^T \left((F - \rho G) \otimes \Sigma^{-1}\right)(\tilde{\boldsymbol{w}} - \tilde{\boldsymbol{m}}) = \sum_{i,j=1}^{I} (F - \rho G)_{ij} (\tilde{\boldsymbol{w}}_i - \tilde{\boldsymbol{m}}_i)^T \Sigma^{-1} (\tilde{\boldsymbol{w}}_j - \tilde{\boldsymbol{m}}_j)$$

  By exploiting multiple times the linearity of the trace operator and its cyclic property, the scale matrix V_p can be seen to equal

$$V_p = \sum_{i,j=1}^{I} (F - \rho G)_{ij} (\tilde{\boldsymbol{w}}_j - \tilde{\boldsymbol{m}}_j)(\tilde{\boldsymbol{w}}_i - \tilde{\boldsymbol{m}}_i)^T + V$$

  and we conclude that $\Sigma \,|\, rest \sim \text{Inv-Wishart}(\nu_p, V_p)$

- Sample ρ from its full conditional:

$$\mathcal{L}(\rho \,|\, rest) \propto \pi(\rho)\mathcal{N}(vec(\tilde{w}_1, \dots, \tilde{w}_I) \,|\, \mathbf{0}, (F - \rho G)^{-1} \otimes \Sigma)$$

This distribution does not have a closed form analytic expression because the support of ρ is $(0, 1)$ and hence we resort to a Metropolis Hastings step. The proposal distribution is a truncated normal (with support on $(0, 1)$) centered in the current value of ρ with standard deviation 0.1. Sampling from the truncated normal is performed by rejection sampling, whereas the computation of the acceptance rate for the Metropolis Hastings step is obtained by exploiting the law of the matrix normal distribution, which does not require to factorize the matrix $(F - \rho G)^{-1} \otimes \Sigma$. To improve the mixing of the chain, we resort to an Adaptive Metropolis Hastings move as in Roberts and Rosenthal (2009) to automatically tune variance of the normal proposal distribution.

- For each $i = 1, \dots, I$ and each $h = 1, \dots H$, independently sample $\tilde{w}_{ih}$ as follows:
  - Sample the latent variable ω_{ih} from

$$\mathcal{L}(\omega_{ih} \,|\, \tilde{\boldsymbol{w}}_i) = PG(N_i, \eta_{ih}) = PG\left(N_i, \tilde{w}_{ih} - \log \sum_{k \neq h} e^{\tilde{w}_{ik}}\right)$$

  - Sample the transformed weight $\tilde{w}_{ih}$ from

$$\mathcal{L}(\tilde{w}_{ih} \,|\, \tilde{\boldsymbol{w}}_{-i}, \tilde{\boldsymbol{w}}_{i,-h}, \boldsymbol{s}_i, \rho, \Sigma, \omega_{ih}) = N(\hat{\mu}_{ih}, \hat{\Sigma}_{ih}).$$

- for each connected component m of the graph we sample from

$$\mathcal{L}(\tilde{\boldsymbol{m}}_{C_m} \,|\, rest) = \mathcal{N}(\boldsymbol{m}_{C_m}, \Lambda_{C_m})$$

For ease of notation, we show how to obtain expression of $\boldsymbol{m}_{C_m}$ and Λ_{C_m} in the case where is only one connected component in the graph. However the general update can be straightforwardly recovered since $\tilde{m}_{C_1}, \dots \tilde{\boldsymbol{m}}_{C_k}$ corresponding to connected components in the graph are conditionally independent a priori. In case of one single connected component in the graph, we rewrite (7.5), letting all the $\tilde{\boldsymbol{m}}_i$s to be equal to $\tilde{\boldsymbol{m}}_1$, as

$$\tilde{\boldsymbol{w}} \sim \mathcal{N}_{I(H-1)}\left(\mathbf{1}_I \otimes \mathbb{I}_{H-1}\tilde{\boldsymbol{m}}_1, \left((F - \rho G) \otimes \Sigma^{-1}\right)^{-1}\right)$$

where $\mathbf{1}_I$ is the vector of ones of length I and $\mathbb{I}_{H-1}$ is the $(H-1) \times (H-1)$ identity matrix. Then if $\Lambda := diag(\sigma^2, \dots, \sigma^2)$ and writing $\mathbb{I}^* = \mathbf{1}_I \otimes I_{H-1}$, $Q = (F - \rho G) \otimes \Sigma^{-1}$, we can write the full conditional of $\tilde{\boldsymbol{m}}_1$ as follows:

$$\begin{aligned}\mathcal{L}(\tilde{\boldsymbol{m}}_1 \,|\, rest) &\propto \exp\left(-0.5\big(\tilde{\boldsymbol{w}} - \mathbb{I}^*\tilde{\boldsymbol{m}}_1\big)^T Q(\tilde{\boldsymbol{w}} - \mathbb{I}^*\tilde{\boldsymbol{m}}_1)^T + \tilde{\boldsymbol{m}}_1^T \Lambda^{-1}\tilde{\boldsymbol{m}}_1\big)\right) \\ &\propto \exp\left(-0.5\tilde{\boldsymbol{m}}_1^T\left(\mathbb{I}^{*T}Q\mathbb{I}^*\right)\tilde{\boldsymbol{m}} + \tilde{\boldsymbol{m}}_1^T\Lambda^{-1}\tilde{\boldsymbol{m}}_1 + -2\tilde{\boldsymbol{m}}_1^T\left(\mathbb{I}^{*T}Q\tilde{\boldsymbol{w}}\right)\right)\end{aligned}$$

This is the kernel of a multivariate normal distribution with covariance matrix $\Lambda_C = \left(\mathbb{I}^{*T}Q\mathbb{I}^* + \Lambda^{-1}\right)^{-1}$ and mean $\boldsymbol{m}_C = \Lambda_C\left(\mathbb{I}^{*T}Q\tilde{\boldsymbol{w}}\right)$.

- If there are covariates in the model $M1$ as in Section 7.7, the full-conditional of the regression coefficients $\boldsymbol{\beta}$ is given by

$$\mathcal{L}(\boldsymbol{\beta} \,|\, rest) = \mathcal{N}_d\left((\Sigma^{-1} + X^TVX)^{-1}(X^TV(\boldsymbol{y} - \boldsymbol{\mu})), (\Sigma^{-1} + X^TVX)^{-1}\right)$$

where V is an $N \times N$ diagonal matrix and $N = \sum N_i$. Denoting by $\boldsymbol{c} = (c_1, \dots, c_N)$, the vectorization of the sequence of latent vectors $\boldsymbol{s}_1, \dots, \boldsymbol{s}_I$ in (7.12)-(7.13), then one has $V_{k,k} = \sigma^2_{c_k}$. The formula above can be derived by standard posterior updates in the Bayesian linear regression, when the mixture model (7.7)-(7.11) is the model for the 'regression error'.

In case of model $M2$, as in Section 7.7, the full-conditional of each regression coefficients $\big((\mu_h, \boldsymbol{\beta}_h), \sigma^2_h\big)$ is straightforwardly computed from standard Bayesian linear regression, considering only observations that are allocated to component h. In particular, if we denote by $\boldsymbol{y}_h$ the vector of $\{y_{ij} : s_{ij} = h\}$, by X_h the matrix with rows $\{x_{ij} : s_{ij} = h\}$, and by n_h the size of $\boldsymbol{y}_h$, then we have

$$\begin{aligned}\sigma^2_h \,|\, rest &\sim IG(a_{ph}, b_{ph}) \\ (\mu_h, \beta_h) \,|\, \sigma^2_h, rest &\sim \mathcal{N}(\boldsymbol{\mu}_{ph}, \Delta_{ph})\end{aligned}$$

where

$$\begin{aligned}\Delta_{ph} &= X_h^T X_h + 10\mathbb{I}_{d+1} \\ \boldsymbol{\mu}_{ph} &= \Delta_{ph}^{-1} X_h^T \boldsymbol{y}_h \\ a_{ph} &= 2 + n_h/2 \\ b_{ph} &= 2 + \frac{1}{2}(\boldsymbol{y}_h^T \boldsymbol{y}_h - \boldsymbol{\mu}_{ph}^T \Delta_{ph} \boldsymbol{\mu}_{ph})\end{aligned}$$

7.D Additional plots and tables

- Figure 7.D.1 shows the total variation distance for $(\boldsymbol{w}_1, \boldsymbol{w}_2)$ and $(\boldsymbol{w}_1, \boldsymbol{w}_4)$ under the logisticMCAR and the prior in Jo et al. (2017) with parameters as in Section 7.4.3. Observe how the distance between $(\boldsymbol{w}_1, \boldsymbol{w}_2)$ decreases as the sparsity increases under both priors. This is expected since areas 1 and 2 are neighbors. However, the distance between $(\boldsymbol{w}_1, \boldsymbol{w}_4)$ increases with sparsity under the logisticMCAR but decreases under CK-SSM. Hence, under CK-SSM, forcing sparsity in the mixture model results in imposing similar behaviors to different connected components.

- Figure 7.D.2 shows draws from the prior mixture density corresponding to parameters sampled from the prior under the logisticMCAR and CK-SSM, having fixed the atoms to have means $\mu_1 = -5$, $\mu_2 = -3.33, \dots, \mu_6 = 5$ and equal variances 0.25^2 and remaining hyperparameters as in Section 7.4.3. It is clear that the logisticMCAR prior allows great variety among disconnected components as well as across different independent samples. Instead, CK-SSM shows that only the first 2/3 components have a nonzero weight, so that the densities across different areas and coming from independent samples are also similar.

- Table 7.D.1 shows the Hellinger distance between the true density and the estimate under the three models under comparison in Section 7.6.1 for the three simulated scenarios in Table 7.6.1 for 100 repeatedly simulated datasets. We average these values over the simulated datasets, also considering ± one empirical standard deviation of the 100 values obtained.

- Figure 7.D.3 shows errors, measured with the Hellinger distance, under our model (spmix) and the HDP-mixture model (hdp) for each simulation, averaged over the areas, for $I = 4, 64, 256$, in Section 7.6.2.

- Figure 7.D.4 displays empirical correlations among the predictors and, in the last column, between predictors and the response for the Airbnb Amsterdam dataset in Section 7.7.

- Figure 7.D.5 shows the scatterplots of the response price versus numerical predictors and boxplots for categorical predictors for the Airbnb Amsterdam dataset in Section 7.7.
- Figure 7.D.6 shows the predictive densities in area Bijlmer-Centrum, corresponding to different covariate specifications: all the covariates are fixed to their empirical median except for `reviews_scores_rating`, which assumes values equal to the empirical quartiles $q_{0.05}, q_{0.5}, q_{0.95}$. It is clear that the three densities overlap almost perfectly. There are two reasons for this. First, the the empirical distribution of this covariate is it is highly concentrated around high values, as people tend to give mostly positive reviews. Second, the coefficient associated to this covariate, despite significant, has a very small absolute value.
- Table 7.7.2 shows the posterior predictive probability $P(y^\star > 200 \,|\, x^\star, i)$ for the same neighborhoods and values of $x^\star$ considered in Figure 7.7.2 (c).

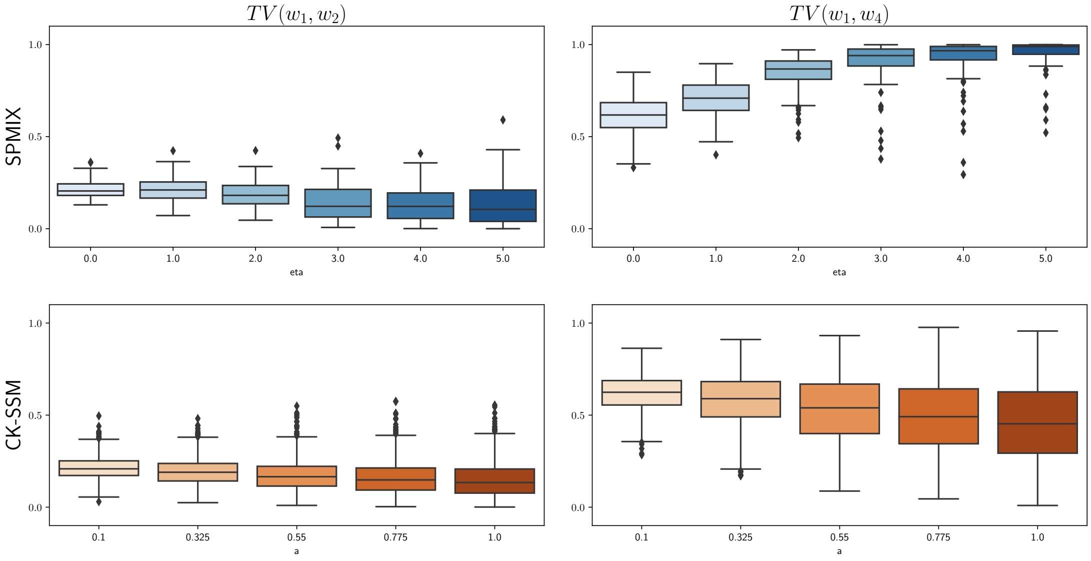


Figure 7.D.1: Total variation distances between the vectors $(\boldsymbol{w}_1, \boldsymbol{w}_2)$ and $(\boldsymbol{w}_1, \boldsymbol{w}_4)$ under the logisticMCAR distribution (first row) and the CK-SSM(second row). Each plot shows the boxplots of 1,000 independents simulations, for different values of the sparsity-tuning parameters (sparsity is increasing from left to right in each plot). The remaining hyperparameters and the adjacency matrix are as discussed in Section 7.4.3.

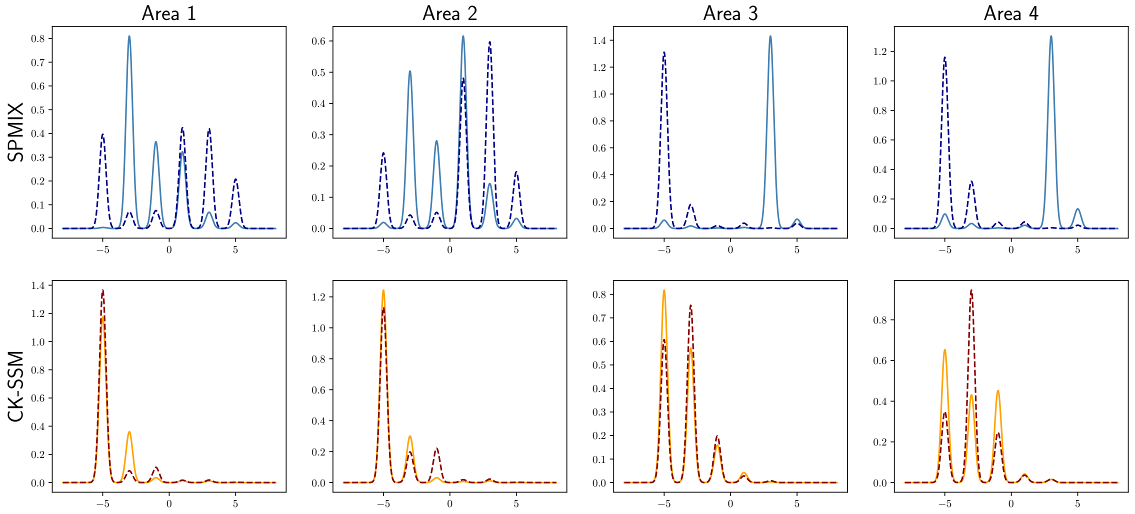


Figure 7.D.2: Samples from the prior distribution with $H = 6$ under the logisticMCAR (first row, with $\eta = 5.0$) and CK-SSM in Jo et al. (2017) (second row, with $b = 0.5, a = 1.0$). The remaining hyperparameters and the adjacency matrix are as discussed in Section 7.4.3. Each plot shows the mixture density in one particular area and different line types / colors represent independent draws from the prior. Here the means of the mixture model are fixed as $\mu_1 = -5$, $\mu_2 = -3.33, \ldots, \mu_6 = 5$ and the variances are all equal to 0.25^2.

	Model	1	2	3	4	5	6
Scenario I	SPMIX	0.06 ± 0.01	0.06 ± 0.01	0.06 ± 0.01	0.06 ± 0.01	0.09 ± 0.01	0.09 ± 0.01
	HDP	0.03 ± 0.01	0.03 ± 0.01	0.06 ± 0.01	0.06 ± 0.01	0.09 ± 0.01	0.09 ± 0.01
	CK-SSM	0.44 ± 0.06	0.44 ± 0.06	0.53 ± 0.03	0.53 ± 0.03	0.44 ± 0.03	0.44 ± 0.03
Scenario II	SPMIX	0.08 ± 0.01	0.11 ± 0.02	0.07 ± 0.01	0.08 ± 0.03	0.11 ± 0.00	0.11 ± 0.03
	HDP	0.04 ± 0.01	0.19 ± 0.02	0.09 ± 0.01	0.24 ± 0.03	0.10 ± 0.00	0.27 ± 0.03
	CK-SSM	0.44 ± 0.06	0.43 ± 0.06	0.53 ± 0.03	0.53 ± 0.03	0.45 ± 0.05	0.45 ± 0.05
Scenario III	SPMIX	0.20 ± 0.07	0.20 ± 0.07	0.16 ± 0.06	0.16 ± 0.06	0.11 ± 0.05	0.11 ± 0.05
	HDP	0.12 ± 0.07	0.12 ± 0.07	0.21 ± 0.06	0.21 ± 0.06	0.13 ± 0.05	0.13 ± 0.05
	CK-SSM	0.42 ± 0.06	0.42 ± 0.06	0.59 ± 0.03	0.59 ± 0.03	0.38 ± 0.07	0.38 ± 0.07

Table 7.D.1: Hellinger distances between the true densities and the estimated ones, aggregated over 100 simulated datasets with ± one standard deviation for the simulated data in Section 7.6.1

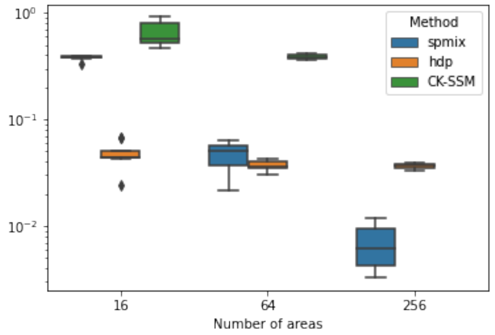


Figure 7.D.3: Boxplots of the Hellinger distance between true density (7.22) and estimated one under our model (spmix), the HDP-mixture model (hdp) and the CK-SSM for each simulation, averaged over the areas, for $I = 16, 64, 256,$, in logarithmic scale, in Section 7.6.2.

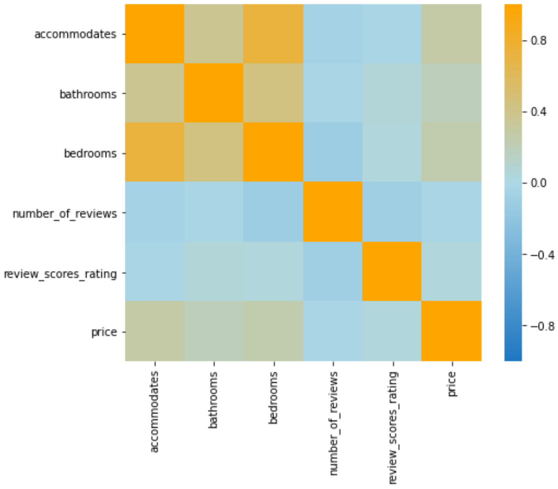


Figure 7.D.4: Correlation matrix between numerical predictors and response for Airbnb Amsterdam

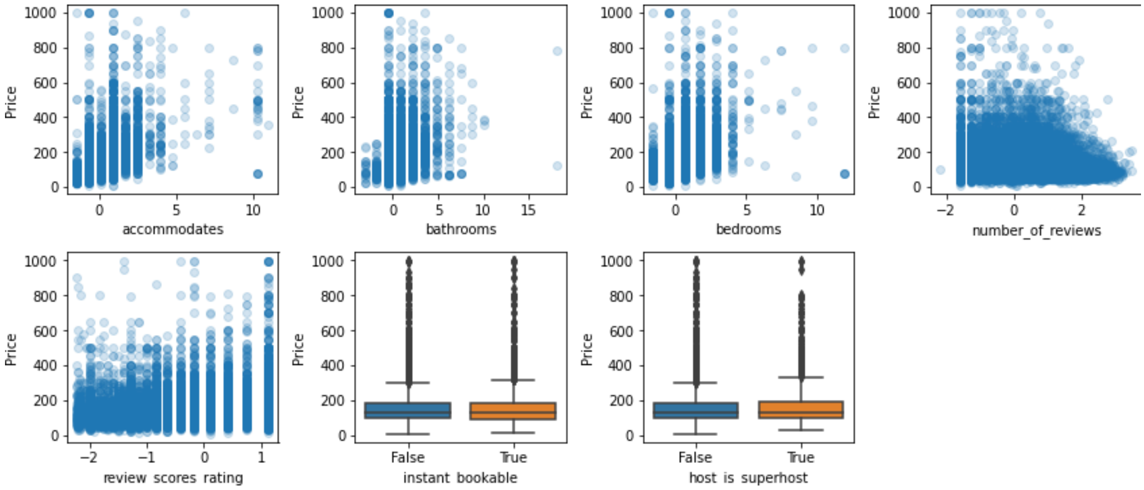


Figure 7.D.5: Scatterplots and boxplots of the nightly price versus predictors for Airbnb Amsterdam. Numerical predictors have been standardized.

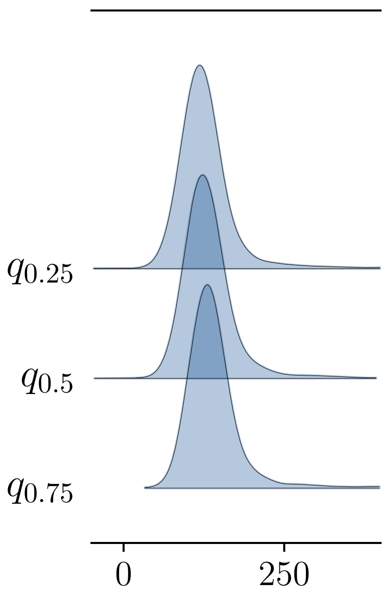


Figure 7.D.6: Predictive density for a new listing in *Bijlmer-Centrum* with all numerical covariates fixed to the empirical median of the dataset except `reviews_scores_rating` that ranges in the values $q_{0.25}, q_{0.5}, q_{0.75}$, where q_α denotes the empirical quantile of order α. Each line corresponds to one of these values, from top to bottom.

7.E Further Comparisons on the Airbnb Amsterdam Dataset

Below, we fit two additional models on the Airbnb dataset, and compare them with our proposal.

7.E.1 Modeling disconnected components separately

As pointed out by one of the reviewers, modeling disconnected components jointly might result in a less flexible model since the parameters ρ and Σ are shared across the spatial components. On the other hand, modeling different connected components independently might result in an overparametrized model with poorer predictive performance.

We compare model $M1$ in Section 7.7 with another model obtained by applying $M1$ separately to the two connected components. By computing LPML and WAIC, we observe that this second model results in a worse predictive error (with a decrease of LMPL and an increase of WAIC of about 0.1 %, for all the values of H considered), showing that the joint estimation of the shared parameters across the disjoint connected components, such as ρ, Σ and τ_h, is beneficial in this scenario.

7.E.2 Geographically Weighted Regression

Another comparison was performed by fitting geographically weighted regression (GWR, Brunsdon et al., 1998) to this dataset. Observe that while our model estimates an entire predictive distribution for each observation, from which different point estimates can be easily derived as discussed in Section 7.7, GWR is a frequentist model that can only provide a point estimate for the mean of the response.

GWR considers observations $\{(y_j, \boldsymbol{x}_j)\}_{j=1}^J$ associated to spatial locations s_j. In our case, the observations are the nightly price (y_j) and the covaraites $(\boldsymbol{x}_j \in \mathbb{R}^d)$ of an Airbnb listing, while the spatial locations are the neighborhoods. The GWR model is then

$$y_j = \boldsymbol{\beta}_{s_j}^t \boldsymbol{x}_j + \varepsilon_j$$

We can assume that the ε_j's are independent and have mean equal to zero. The coefficients $\boldsymbol{\beta}_{s_j}$ (observe that in our case there is one for each neighborhood), are estimated by solving a weighted least-squares optimization problem so that:

$$\boldsymbol{\beta}_{s_j} = \left(X^T W_{s(y_j)} X\right)^{-1} \left(X^T W_{s_j} Y\right)$$

where X is the $J \times d$ matrix with rows x_j, Y is the vector with entries y_j and W_{s_j} is a diagonal matrix whose elements are a function of the distance between location s_j and the location associated to observation y_i. With an abuse of notation, denote with $s(y_i)$ the neighborhood observation y_i belongs to, then $(W_{s_j})_{ii} = g(d(s_j, s(y_i)))$.

As in Brunsdon et al. (1998), we let the weights $g(d(s_j, s(y_i))) = \exp(-\gamma d(s_j, s(y_i)))$. For two neighborhoods s_i and s_j we let $d(s_i, s_j)$ be the the shortest path distance on the graph G between neighborhood i and the neighborhood which observation j belongs to.

Performance is measured by means of pMSE through a 10-fold cross-validation. In each fold, γ is chosen among the values $\{0.01, 0.1, 0.5, 1.0, 2.0, 5.0, 10.0\}$ via another (nested) cross-validation.

The resulting pMSE is much higher (roughly five times) than the one from $M1$, for all values of H. While GWR results to be not competitive with our model in this case, we expect that it might yield better performances in other scenarios with a finer spatial scale (e.g., geolocation of each observation). In such cases, GWR could be preferred to provide a point estimate.

8. Normalized latent factor measure models

In this chapter, based on Beraha and Griffin (2022), we propose a methodology for modeling and comparing probability distributions within a Bayesian nonparametric framework. Building on dependent normalized random measures, we consider a prior distribution for a collection of discrete random measures where each measure is a linear combination of a set of *latent* measures, interpretable as characteristic traits shared by different distributions, with positive random weights. The model is non-identified and a method for post-processing posterior samples to achieve identified inference is developed. This uses Riemannian optimization to solve a non-trivial optimization problem over a Lie group of matrices. The effectiveness of our approach is validated on simulated data and in two applications to two real-world data sets: school student test scores and personal incomes in California. Our approach leads to interesting insights for populations and easily interpretable posterior inference.

8.1 Introduction

Modeling a set of related probability measures is a common task in Bayesian statistics, the most common example being when covariates are associated with each observation. In this work, we consider the case of a single discrete-valued covariate, which might be regarded as a group indicator, that is, when data are naturally divided into subpopulations or groups. One of the main motivations for these kinds of analyses is combining data from different sources or experiments, where, for each source, a set of observations is collected: pooling together all the data could ignore important differences across populations while modeling each group separately might result in poor performance especially if the number of observations in each group is small. Applications range from population genetics (Elliott et al., 2019) to healthcare (Müller et al., 2004; Rodriguez et al., 2008) and text mining (Teh et al., 2006).

Within this setting, our goal is to propose a flexible model that, in addition to combining heterogeneous sources of data, gives an efficient way of representing the difference in distribution across populations. Consider for example Figure 8.1.1, which displays the distribution of the personal annual income (on the log scale) in four different geographic areas of California: two in Los Angeles and two in San Francisco. In this case, similarities and differences between the distributions can be easily spotted by eye: the two areas in Los Angeles are associated with (much) lower incomes than the areas in San Francisco. When the number of groups increases, it is not possible to carry out these comparisons by eye. Our model provides a way to decompose the area-specific densities into a linear combination of "common traits", which are themselves probability measures. In Section 8.6.2, we provide a thorough analysis of the Californian income data, finding four common traits, associated with an average distribution of income, and a prevalence of low, medium, and high incomes respectively. By looking at the weights (of the linear combination of common traits) associated with the four groups in Figure 8.1.1, we easily spot differences between the Los Angeles and San Francisco areas: the weight associated to the low-income trait is large in the first areas and low in the second two; vice versa for the weight associated to

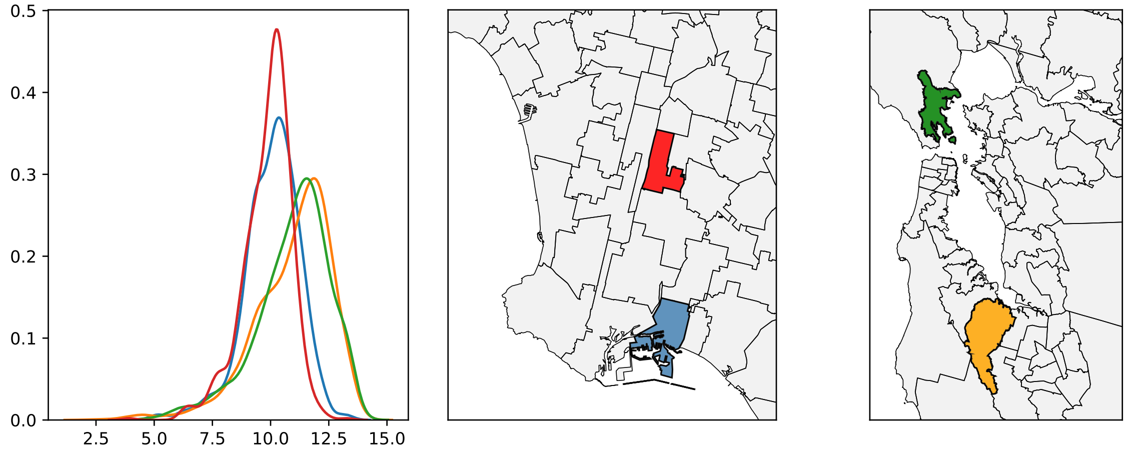


Figure 8.1.1: Kernel density estimates of the (log) personal incomes in four areas in California (left plot): two in Los Angeles (middle plot) and two in San Francisco (right plot).

the high-income trait. See Figure 8.6.3 for more details.

To formalize the discussion above, let $(\boldsymbol{y}_1, \dots, \boldsymbol{y}_g)$, $\boldsymbol{y}_j = (y_{j1}, \dots, y_{jn_j})$ denote a sample of observations divided into g groups. A common assumption is that data are exchangeable in each group, but exchangeability might not hold across different groups. In particular, by de Finetti's theorem, this is tantamount to assuming that there is a vector of random probability measures $(p_1, \dots, p_g) \sim Q$ such that, in each group, $y_{j1}, \dots, y_{jn_j} \overset{\text{iid}}{\sim} p_j$ and that independence, conditionally on $p_1, \dots, p_g$, holds across groups. We focus here on mixture models of the kind $p_j(y) = \int_\Theta f(y \mid \theta)\tilde{p}_j(\mathrm{d}\theta)$.

The construction of a flexible prior Q that can suitably model heterogeneity while borrowing information across different groups has been thoroughly studied in Bayesian nonparametrics. Previously proposed approaches consider constructing $\tilde{p}_1, \dots, \tilde{p}_g$ in a hierarchical model fashion (Teh et al., 2006; Camerlenghi et al., 2019; Bassetti et al., 2020; Argiento et al., 2019; Beraha et al., 2021), considering convex combinations of shared and group-specific random measures (Müller et al., 2004), starting from additive processes (Griffin et al., 2013; Lijoi et al., 2014a) and nested processes (Rodriguez et al., 2008; Camerlenghi et al., 2019). See Quintana et al. (2022) for a recent review.

As previously mentioned, the focus of the present chapter is slightly different. First of all, we are interested in the situation when the number of groups g is large relative to the sample size in each group n_j. Then, it is likely that the dataset cannot inform the huge number of parameters that are associated with extremely flexible models and we advocate for a more parsimonious model where substantial sharing of information is encouraged across different groups of data. Moreover, in addition to modeling the densities $\tilde{p}_1, \dots, \tilde{p}_g$, we also want to identify the main differences in distribution of the data across groups. To the best of our knowledge, this question has not been addressed systematically in the Bayesian nonparametric literature. In the frequentist one, several approaches to principal component analysis for probability distribution have been proposed, see for instance Pegoraro and Beraha (2022) and the references therein.

The setting "large g, small n_j" is somewhat reminiscent of high-dimensional data analysis, where the dimension of each observation is large relative to the sample size. In this case, latent factor models (see, e.g., Arminger and Muthén, 1998) provide a powerful tool. In a latent factor model, it is assumed that each observation $x_i \in \mathbb{R}^p$ is a linear combination of a set of H d-dimensional latent factors weighted by observation-specific scores, plus an isotropic error term. We follow this analogy and propose *normalized latent measure* *factor models*, a class of prior distributions for a vector of random probability measures $\tilde{p}_1, \ldots, \tilde{p}_g$. Informally, our model amounts to considering $\tilde{p}_j$ as a convex combination of a set of latent random probability measures, see Section 8.8.

Our construction shares similarities with Griffin et al. (2013) and Lijoi et al. (2014a). There, the authors assume each $\tilde{p}_j$ as the normalization of a random measure obtained by superposing several completely random measures. Essentially, this is analogous to our approach if we let all the scores (before some normalization step, see Section 8.8) be zero or one. The main difference is that, since their scores are binary, they usually assume that the number of latent factors H is larger than the number of groups g. This leads to posterior simulation algorithms that can scale and/or mix poorly with g. Moreover, they do not consider the problem of decomposing the populations' distribution into interpretable common traits, which necessarily requires H to be much smaller than g.

As is usually the case for latent factor models, our model is not identifiable. To tackle this issue, we propose post-processing the MCMC chains o find an "optimal representative" for both the latent factor loadings matrix and the latent random measures which leads to a non-trivial optimization problem. Indeed, taking into account the invariance to scaling of normalized random measures leads to formulating the optimization over a Riemannian manifold of matrices, specifically the special linear group (matrices whose determinant is equal to one). Moreover, additional constraints must be taken into account to ensure the positiveness of the loadings matrix and latent random measures. We propose an iterative algorithm based on gradient descent. The first constraint (determinant equal to one) can be tackled by means of differential geometric tools: leveraging the differential structure of the special linear group, we use a variant of Riemannian gradient descent which ensures that all the intermediate points of the algorithm lie inside the special linear group. To take into account the positivity constraints, we propose to use the augmented Lagrangian multiplier method within the previously discussed Riemannian framework, leading to a Riemannian augmented Lagrangian multiplier method.

We consider two motivating applications. The first one is the scores on a mathematics test of approximately $40,000$ students in 1048 Italian high schools from the *invalsi* dataset. The median number of students taking the test in each high school is as little as 37, the minimum being 4 and the maximum 131. The second one comes from the US income survey. Here, the groups are represented by geographical units called *PUMAs*, which correspond to areas with roughly $100,000$ inhabitants. We show how our model can be adapted to induce correlation between PUMAs that are geographically close, by assuming that the scores are distributed as a log Gaussian Markov random field. Compared to traditional spatial factor models, we introduce the spatial dependence in the loadings matrix instead of the latent factors.

The rest of the chapter is organized as follows. Section 8.8 formalizes our model and discusses its statistical properties. Section 8.3 describes the MCMC algorithm for posterior inference and we present our post-processing algorithm in Section 8.4. Section 8.5 and Section 8.6 present numerical illustration on simulated data and real data, respectively. Finally, we discuss possible extensions of the proposed approach in Section 8.7. The Appendix collects background material on Riemannian optimization and completely random measures, proofs of the theoretical results, and additional simulations. `Python` code implementing the MCMC and the post-processing algorithms is available at github.com/mberaha/nrmifactors.

8.2 The Model

For simplicity and specifity, we assume that each $y_{ji} \in \mathbb{R}^d$ and that $\Theta \subset \mathbb{R}^q$ for some d, q. The results can be easily extended to the case when y_{ji} are elements of a complete and separable (i.e., Polish) metric space and Θ is Polish as well.

To keep the discussion light, we defer all technical details and the proofs of the results to the Appendix.

8.2.1 Preliminaries

Before presenting our model in detail, we give some background material on completely random measure and their normalization. This will constitute the backbone of our approach.

Let $(\Theta, \mathcal{B}(\Theta))$ be a complete and separable metric space endowed with its Borel σ-algebra. A random measure is a random element μ taking values in the space of probability measures over Θ, such that $\mu(B) < +\infty$ almost surely for all $B \in \mathcal{B}(\Theta)$. Such a measure is termed completely random by Kingman (1967) if, for pairwise disjoint $B_1, \ldots, B_n \in \mathcal{B}(\Theta)$, the random variables $\mu(B_j)$, $j = 1, \ldots, n$, are independent. For our purposes, it is sufficient to consider completely random measures of the kind $\mu(A) = \int_{\mathbb{R}_+ \times A} sN(\mathrm{d}s\,\mathrm{d}x)$, where N is a Poisson point process on $\Theta \times \mathbb{R}_+$ with base (intensity) measure $\rho(\mathrm{d}s\,\mathrm{d}x)$. We further assume $\rho(\mathrm{d}s\,\mathrm{d}x) = \nu(\mathrm{d}s)\,\alpha(\mathrm{d}x)$ where ν is a Lévy measure on the positive reals, α is a Borel measure on Θ. See, e.g., Kingman (1993) for a detailed account of random measures.

A fruitful approach to constructing random probability measures is by normalization of completely random measures, i.e., by setting $p(\cdot) = \mu(\cdot)/\mu(\Theta)$, which was originally introduced in Regazzini et al. (2003). For the random measure p to be well defined, one must ensure that $\mu(\Theta) > 0$ and $\mu(\Theta) < +\infty$ almost surely. As shown in Regazzini et al. (2003), sufficient conditions are $\int_{\mathbb{R}_+} \nu(\mathrm{d}s) = +\infty$ and $\int_{\mathbb{R}_+} \min\{1, s\}\,\nu(\mathrm{d}s) < +\infty$.

8.2.2 Normalized Latent Measure Factor Models

As already mentioned in the Introduction, we assume

$$y_{j1}, \ldots, y_{jn_j} \,|\, \tilde{p}_j \overset{\text{iid}}{\sim} p_j := \int_\Theta f(\cdot \,|\, \theta)\tilde{p}_j(\mathrm{d}\theta)$$

and that each $\tilde{p}_j$ is a normalized random measure, that is

$$\tilde{p}_j(\cdot) = \frac{\widetilde{\mu}_j(\cdot)}{\widetilde{\mu}(\Theta)}, \qquad j = 1, \ldots, g.$$

Then, the model is specified by a choice of the mixture kernel $f(\cdot \,|\, \cdot)$ and a prior distribution for $(\widetilde{\mu}_1, \ldots, \widetilde{\mu}_g)$. Let $\mu_1^*, \ldots, \mu_H^*$ be a completely random vector (i.e., a vector of completely random measures). Let λ_{jh}, $j = 1, \ldots, g$, $h = 1, \ldots, H$ be a double sequence of almost surely positive random variables (specific choices of the distribution of the λ_{jh}'s are discussed later). We assume

$$\widetilde{\mu}_j(\cdot) = \sum_{h=1}^{H} \lambda_{jh}\,\mu_h^*(\cdot). \tag{8.1}$$

We could choose $(\mu_1^*, \ldots, \mu_H^*)$ to be independent and identically distributed random measures, i.e.

$$\mu_h^*(\cdot) = \sum_{k \geq 1} W_{hk}\,\delta_{\theta_{hk}^*}(\cdot)$$

where $\{W_{hk}, \theta_{hk}^*\}_{k=1}^{\infty}$ are the points of a Poisson point process on $[0, +\infty) \times \Theta$ with, for instance, intensity $\nu_h(\mathrm{d}s_h\,\mathrm{d}x_h) = \rho(s_h)\mathrm{d}s_h\,\alpha(\mathrm{d}x_h)$, i.e., all the intensities are equal. This choice leads to a particularly tractable model for $(\widetilde{\mu}_1, \ldots, \widetilde{\mu}_g)$ as we have that marginally, each $\widetilde{\mu}_j$ is a completely random measure as specified in the following proposition.

Proposition 8.1. *Let $\widetilde{\mu}_j = \sum_{h=1}^H \lambda_{jh}\mu_h^*$ where the μ_h^*'s are completely random measures with associated Lévy intensity $\nu_h^*(\mathrm{d}s_h, \mathrm{d}x_h) = \rho_h^*(s_h)\mathrm{d}s_h\, \alpha_h^*(\mathrm{d}x_h)$. Further, assume that the μ_h^*'s are independent. Then $\widetilde{\mu}_j$ is a completely random measure with Lévy intensity*

$$\nu_j(\mathrm{d}s, \mathrm{d}x) = \sum_{h=1}^H \frac{1}{\lambda_{jh}} \rho_h^*(s/\lambda_{jh})\alpha_h^*(\mathrm{d}x)$$

We find that a more suitable model for our applications arises when $\mu_1^*, \ldots, \mu_H^*$ share their support points. In particular, we will assume that $\mu_1^*, \ldots, \mu_H^*$ is a compound random measure (CoRM, Griffin and Leisen, 2017). That is,

$$\mu_h^*(\cdot) = \sum_{k\geq 1} m_{hk} J_k \delta_{\theta_k^*}(\cdot),$$

where m_{hk} are positive random variables such that $m_k = (m_{1k}, \ldots, m_{Hk})$, $k \geq 1$, are independent and identically distributed from a probability measure on $\mathbb{R}_+^H$, and $\eta = \sum_{k\geq 1} J_k\delta_{\theta_k^*}$ is a completely random measure with Lévy intensity $\nu^*(\mathrm{d}z)\alpha(\mathrm{d}x)$. We argue that a CoRM-based construction should be preferred to an independent CRMs-based one since (i) sharing atoms across all measures is linked to better predictive performance (Quintana et al., 2022), (ii) the number of parameters involved is much smaller, which ultimately leads to the possibility of fitting this model to large datasets, and (iii) each latent factor μ_h^* can be interpreted separately (through the post-processing algorithm presented in Section 8.4). The effectiveness of this model comes with a tradeoff in analytical tractability, since, as shown in the Appendix, the random measure (8.2) is not completely random.

In this case we can write

$$\widetilde{\mu}_j(\cdot) = \sum_{k\geq 1} (\Lambda M)_{jk} J_k \delta_{\theta_k^*}(\cdot), \tag{8.2}$$

where Λ is the $J \times H$ matrix with entries λ_{jh}, M is a $H \times \infty$ matrix, so that $\Gamma = \Lambda M$ is a $g \times \infty$ matrix with entries γ_{jk}, $j = 1, \ldots, g$, $k \geq 1$. Note that, in analogy to CoRMs, also our model includes shared weights J_k for all the measures $\widetilde{\mu}_j$. We find that the additional borrowing of strength obtained through the J_k's is useful in practice since, in our applications, the $\widetilde{\mu}_j$'s are usually similar.

Equations (8.1) and (8.2) share analogies to latent factor models, where the observed variable is $X \in \mathbb{R}^p$ and its ℓ-th entry is modeled as $X_\ell \approx \sum_{h=1}^H \omega_{\ell h} Z_h$, for $Z = (Z_1, \ldots, Z_H)$ an H-dimensional random variable. In particular, we could consider $\mu_1^*, \ldots, \mu_H^*$ to be measure-valued factor loadings and the λ_{jh}'s to be factor scores. This yields an interpretation similar to functional factor models (Montagna et al., 2012). On the other hand, we could consider the measure-valued vector $(\widetilde{\mu}_1, \ldots, \widetilde{\mu}_g)$ as a single high-dimensional observation, and model it as a linear combination of measure-valued factors with loadings λ_{jh}'s. Both interpretations make sense and lead to interesting analogies. We use the latter one and call Λ the loadings matrix and the μ_h^*'s the latent measures.

Prior elicitation is required to set the Lévy intensity ν^* of the CoRM, the distribution of the scores m_{hk}, and the distribution of Λ. Following Griffin and Leisen (2017), we assume that $m_{hk} \overset{\text{iid}}{\sim} \mathrm{Ga}(\phi)$, where $\mathrm{Ga}(\phi)$ denotes the law of a gamma random variable with shape parameter ϕ and rate parameter 1(we will also use $\mathrm{Ga}(\phi, \beta)$ to denote a gamma random variable with rate parameter $\beta \neq 1$). Therefore, the dependence across the $\widetilde{\mu}_j$'s depends on H, ν^*, and Λ.

The prior for Λ allows us to address several interesting modeling questions. When no additional group-specific information is available, such as comparing the distribution of test results in different schools, a natural choice would be to assume the λ_{ij}'s i.i.d.

from some probability distribution with support on $\mathbb{R}_+$, such as the gamma distribution. We find it more convenient to specify a *shrinkage* prior on Λ, to automatically select the number of latent factors H. This approach has received considerable attention in Gaussian latent factor models, see, for instance, Bhattacharya and Dunson (2011); Legramanti et al. (2020); Schiavon et al. (2022). In our example, we consider Λ distributed as a multiplicative gamma process (Bhattacharya and Dunson, 2011),

$$\lambda_{jh} = (\phi_{jh}\tau_h)^{-1},\ \tau_h = \prod_{j=1}^{h} \theta_j,\ \theta_1 \sim \mathrm{Ga}(a_1),\ \theta_2, \ldots \overset{\text{iid}}{\sim} \mathrm{Ga}(a_2),\ \phi_{jh} \overset{\text{iid}}{\sim} \mathrm{Ga}(\nu/2, \nu/2). \quad (8.3)$$

In Section 8.3 we propose a variant of the adaptive Gibbs sampler of Bhattacharya and Dunson (2011) to automatically select H in the first iterations of the MCMC algorithm.

If group-specific information, such as covariates, is available, we can model the finite-dimensional matrix Λ. For example, the PUMAs in the Californian income data are indexed by a specific areal location. This can be modelled using a $g \times g$ spatial proximity matrix denoted by W, where $W_{j\ell} = 1$ if areas j and ℓ share an edge and $W_{j\ell} = 0$ otherwise, but more general choices of proximity could be considered in other examples. Then, we can encourage spatial dependence between the $\widetilde{\mu}_j$'s by assuming

$$\log \boldsymbol{\lambda}^h \overset{\text{iid}}{\sim} \mathcal{N}_H\left(\mu, (\tau(F - \rho W))^{-1}\right), \qquad h = 1, \ldots, H \qquad (8.4)$$

where $\boldsymbol{\lambda}^h = (\lambda_{1h}, \ldots, \lambda_{gh})$ is the h–th column of the matrix Λ, F is a diagonal matrix with entries $F_{ii} = \sum_j W_{ij}$, and $\rho \in (0, 1)$. We suggest setting $\mu = \log(1/H, \ldots, 1/H)$ in (8.4) to encourage a priori each $\widetilde{\mu}_j$ to be a convex combination of the μ_h^*'s with equal weights. The model could also be applied to geo-referenced data using a log Gaussian process,

$$\log \boldsymbol{\lambda}^h \overset{\text{iid}}{\sim} \mathcal{GP}(\mu, \mathcal{K}), \qquad h = 1, \ldots, H$$

where $\boldsymbol{\lambda}^h = (\lambda_{1h}, \ldots, \lambda_{gh})$ is the h–th column of the matrix Λ.

8.2.3 Some Statistical Properties

In this section, we discuss some distributional properties of the measures $\widetilde{\mu}_j$'s in light of the prior assumption above. We assume that the λ_{jh}'s are independent of $\mu_1^*, \ldots, \mu_H^*$. Firstly, it is clear that

$$\mathbb{E}[\widetilde{\mu}(A)] = \sum_{h=1}^{H} \mathbb{E}[\lambda_{jh}]\mathbb{E}[\mu_h^*(A)].$$

When we consider the normalized measures, the expression of the expected value is more complex.

Theorem 8.1. *Let $(\mu_1^*, \ldots, \mu_H^*)$ be a CoRM with i.i.d. scores. Denote the Laplace transform of the scores' distribution by $\mathcal{L}_m(u) := \mathbb{E}[e^{-um}]$ and let $\kappa_m(u, n) := \mathbb{E}[e^{-um}m^n]$. Then for all measurable $A \subset \Theta$*

$$\mathbb{E}[\tilde{p}_j(A)] = \\ \alpha(A) \sum_{h=1}^{H} \int \mathbb{E}\left[\lambda_{jh}\psi_\rho(u\lambda_{j1}, \ldots, u\lambda_{jH}) \int_{\mathbb{R}_+} z \prod_{k \neq h} \mathcal{L}_m(u\lambda_{jk}z)\kappa_m(u\lambda_{jh}z, 1)\nu^*(\mathrm{d}z)\right] \mathrm{d}u$$

where ψ_ρ is the Laplace functional of $(\mu_1^*, \ldots, \mu_H^*)$ (evaluated at the constant functions $u\lambda_{j1}, \ldots, u\lambda_{jH}$).

Although it is not possible to evaluate the quantity in Theorem 8.1 analytically, a priori Monte Carlo simulation can be used to numerically estimate the expected value of $\tilde{p}_j(A)$.

To characterize the dependence induced by the latent measure factor model, an intuitive measure is the covariance between two random measures.

Proposition 8.2. *The following expression holds.*

$$\begin{aligned}&Cov\,[\widetilde{\mu}_j(A), \widetilde{\mu}_\ell(B)] = \\ &\qquad \sum_{h,k} \mathbb{E}[\lambda_{jh}\lambda_{\ell k}]\,Cov(\mu^*_h(A), \mu^*_k(B)) + Cov(\lambda_{jh}, \lambda_{\ell k})\mathbb{E}[\mu^*_h(A)\mu^*_k(B)] \qquad (8.5)\end{aligned}$$

If the λ_{jh}'s have the same marginal distribution, the μ^*_h's have the same marginal distribution, $\lambda_j = (\lambda_{j1}, \dots, \lambda_{jH})$ and λ_ℓ (defined analogously) are independent, $\mathbb{E}[\lambda_{jh}\lambda_{\ell h}] = \kappa$, $Cov(\lambda_{jh}, \lambda_{\ell h}) = \rho$ for all j, ℓ, h, then:

$$\begin{aligned}&Cov\,[\widetilde{\mu}_j(A), \widetilde{\mu}_\ell(B)] = \\ &\qquad Cov(\mu^*_1(A), \mu^*_1(B))\kappa H + m^*_1(A)m^*_1(B)\rho H + \sum_{h \neq q} \bar{\lambda}^2_{11}\,Cov(\mu^*_h(A), \mu^*_k(B))\end{aligned}$$

where $\bar{\lambda}_{jh} := \mathbb{E}[\lambda_{jh}]$ and $m^*_h(A) = \mathbb{E}[\mu^*_h(A)]$.

Finally, if in addition the μ^*_h's are independent, the latter sum disappears

From (8.5), it is clear that Cov $[\widetilde{\mu}_j(A), \widetilde{\mu}_\ell(B)]$ increases with: (i) the correlation of the measures at the latent lavel (Cov($\mu^*_h(A), \mu^*_k(B)$) large), (ii) the correlation of the scores (Cov($\lambda_{jh}, \lambda_{\ell k}$) large), (iii) large values in the scores ($\mathbb{E}[\lambda_{jh}\lambda_{\ell k}]$ large), (iv) random measures with large masses ($\mathbb{E}[\mu^*_h(A), \mu^*_k(B)]$ large), and (v) large values of H (more terms in the summation).

The correlation between $\widetilde{\mu}_j(A)$ and $\widetilde{\mu}_\ell(B)$ can be formally derived from (8.5) but its expression is not easily interpretable in general. To get a nicer expression, assume $A = B$, Cov($\mu^*_h(A), \mu^*_k(A)$) = Cov($\mu^*_m(A), \mu^*_n(A)$) = c_A, $\mathbb{E}[\mu^*_h(A)] = \mathbb{E}[\mu^*_k(A)] = m_A$. Then

$$\begin{aligned}\mathrm{Cov}\,[\widetilde{\mu}_j(A), \widetilde{\mu}_\ell(A)] &= \mathbb{E}[\mu^*_1(A)^2]\left(\sum_{h=1}^{H}\mathbb{E}[\lambda_{jh}\lambda_{\ell h}]\right) + \\ &\qquad (c_A + m_A^2)\left(\sum_{h \neq k}\mathbb{E}[\lambda_{jh}\lambda_{\ell k}]\right) - m_A^2\left(\sum_{h,k}\bar{\lambda}_{jh}\bar{\lambda}_{\ell k}\right)\end{aligned}$$

Let us specialize the above expression further. Consider first the case of independent scores $\lambda_{jh} \overset{\text{iid}}{\sim} \mathrm{Ga}(\psi, 1)$. The correlation between $\widetilde{\mu}_j(A)$ and $\widetilde{\mu}_\ell(A)$ amounts to

$$\left(1 + \frac{m_A}{\left(\mathrm{Var}[\mu^*_1(A)] + c_A(H-1)\right)\psi}\right)^{-1} \qquad (8.6)$$

which is an increasing function of H and ψ as expected. See Appendix 8.B for a proof.

To evaluate m_A, and c_A we use the following result.

Proposition 8.3. *Consider a CoRM with Ga(ϕ) distributed scores and gamma process marginals (i.e., each μ^*_h is distributed as a gamma process). Then for any measurable A:*

1. *$\mathbb{E}[\mu^*_h(A)] = \alpha(A)$,*
2. *$\mathbb{E}[\mu^*_h(A)\mu^*_k(A)] = (\alpha(A) + \alpha(A)^2)\phi^2(B(1, \phi))^2 3/2$, where $B(a, b)$ denotes the Beta function.*

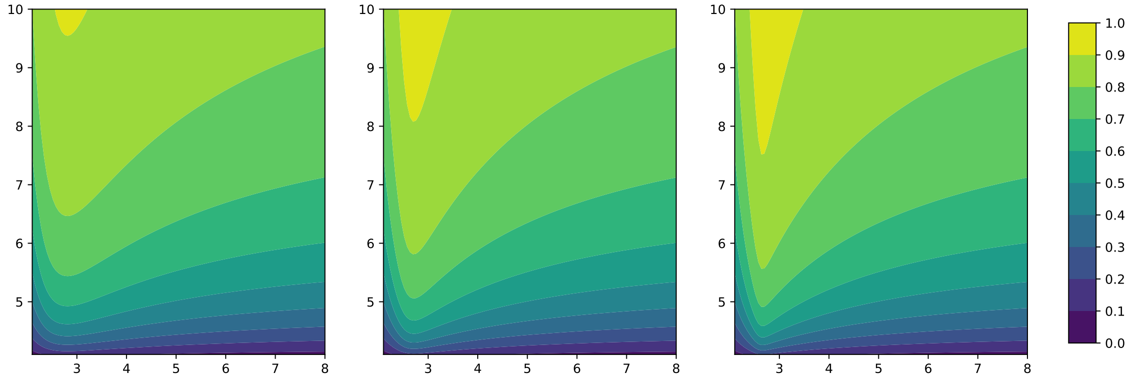


Figure 8.2.1: Correlation between $\widetilde{\mu}_j(A)$ and $\widetilde{\mu}_\ell(A)$ for a set A such that $\alpha(A) = 0.5$, under the multiplicative gamma process prior. $a_1 = 2.5$, $\phi = 2$. From left to right $H = 4, 8, 16$. The values of a_2 vary across the x-axis in each plot, the values of ν across the y-axis.

Consider now the case when Λ is distributed as a multiplicative gamma process introduced in Bhattacharya and Dunson (2011). In this case, we don't have an interpretable expression for the correlation between $\widetilde{\mu}_j(A)$ and $\widetilde{\mu}_\ell(A)$. In the Appendix 8.B we report the expressions for $\mathrm{Cov}\left[\widetilde{\mu}_j(A), \widetilde{\mu}_\ell(A)\right]$ and $\mathrm{Var}[\widetilde{\mu}_j(A)]$ which might be used to numerically compute the desired correlation. Figure 8.2.1 displays the correlation between $\widetilde{\mu}_j(A)$ and $\widetilde{\mu}_\ell(A)$ for a set A such that $\alpha(A) = 0.5$. We notice that when the CoRM has gamma process marginals, the parameter ϕ has little effect on the correlation between the $\widetilde{\mu}_j$'s. On the contrary, there is a strong interaction between a_2, ν, and H. For smaller values of ν, larger values of H imply a higher correlation. When ν is sufficiently large (e.g. larger than 6), the effect of H is less evident. Moreover, larger values of a_2 imply a weaker correlation. This is expected as it essentially reduces the number of active latent measures. In Figure 8.E.1 in the Appendix, we show the correlation between $\widetilde{\mu}_j(A)$ and $\widetilde{\mu}_\ell(A)$ under prior (8.4) for different choices of areas j and ℓ, as a function fo τ and ρ.

Since the atoms are shared across all the measures $\widetilde{\mu}_j$'s, another possible way of characterizing the dependence between two measures is to consider the ratio of weights associated to the k–th atom in $\widetilde{\mu}_j$ and $\widetilde{\mu}_\ell$,

$$r^k_{j\ell} := \frac{(\Lambda M)_{jk}}{(\Lambda M)_{\ell k}} = \frac{\sum_{h=1}^H \lambda_{jh} m_{hk}}{\sum_{h=1}^H \lambda_{\ell h} m_{hk}} \tag{8.7}$$

A trivial upper bound is

$$r^k_{j\ell} \leq \sum_{h=1}^H \frac{\lambda_{jh}}{\lambda_{\ell h}}$$

Multiplying and dividing by H in (8.7) and taking the logarithm yields

$$\log r^k_{j\ell} = \log\left(\frac{1}{H}\sum_{h=1}^H \lambda_{jh} m_{hk}\right) - \log\left(\frac{1}{H}\sum_{h=1}^H \lambda_{\ell h} m_{hk}\right).$$

By the strong law of large numbers, we have that $\log r^k_{j\ell} \to 0$ as $H \to \infty$ if, for instance, λ_{jh} and $\lambda_{\ell h}$ are independent and identically distributed across the values of h. Moreover, it is clear that the variance of $r^k_{j\ell}$ increases with the variance of the λ_{jh}'s. In Appendix 8.E we report an a prior Monte Carlo simulation comparing $r_{j\ell}$ as a function of H under different priors for Λ, namely and i.i.d. prior with $\mathrm{Ga}(\psi)$ distributed λ_{jh}'s, the multiplicative gamma process in (8.3) and the the cumulative shrinkage prior Legramanti et al. (2020).

It is clear that under the two latter shrinkage priors, the choice of H has a smaller impact on the prior. For the sake of computational efficiency, we will adopt the multiplicative gamma process prior in our simulations, when no additional group-specific covariates are present. Instead, when we consider the case of area-referenced groups, we consider H to be a hyperparameter and perform model selection based on predictive performance

8.3 Posterior Inference

Let α be a measure on Θ, ν^* a Lévy intensity on $\mathbb{R}_+$, and $\phi > 0$. We denote with $\mathrm{CoRM}(\phi, \nu^*, \alpha)$ the law of a compound random measure with i.i.d. $\mathrm{Ga}(\phi)$-distributed scores with directing random measure with intensity $\nu^*(z)\mathrm{d}z\,\alpha(\mathrm{d}\theta)$. Our model can be compactly summarized as

$$\begin{aligned} y_{ji} \mid \theta_{ji} &\overset{\text{ind}}{\sim} k(\cdot \mid \theta_{ji}), \qquad i = 1, \ldots, n_i \\ \theta_{ji} \mid \widetilde{\mu}_j &\overset{\text{iid}}{\sim} \widetilde{\mu}_j / \widetilde{\mu}_j(\Theta), \qquad i = 1, \ldots, n_i \\ \widetilde{\mu}_j &:= \sum_{h=1}^{H} \lambda_{jh} \mu^*_h \\ (\mu^*_1, \ldots, \mu^*_h) &\sim \mathrm{CoRM}(\phi, \nu^*, \alpha), \qquad \Lambda \sim \pi(\Lambda) \end{aligned} \tag{8.8}$$

In this section, we describe a simple MCMC scheme based on a truncation of the random measures. In particular, let $K > 0$ denote a fixed number of atoms, we set

$$\mu^*_h = \sum_{k=1}^{K} m_{hk} J_k \delta_{\theta^*_k}$$

where $J_k \overset{\text{iid}}{\sim} p_J$, with p_J being a probability distribution, and $\theta^*_k \overset{\text{iid}}{\sim} G_0 := \alpha/\alpha(\Theta)$. Campbell et al. (2019) provide a thorough review of truncation methods for completely random measures including the choice of p_J for different random measures. We use $p_J = \mathrm{Beta}(\phi/K, \phi)$ so that $\sum_{k=1}^{K} J_k \delta_{\theta^*_k}$ converges to a Beta process as $K \to +\infty$. This combined with gamma-distributed m_{hk} imply that marginally μ^*_h follows a gamma process (see Griffin and Leisen, 2017). Although this simple truncation might result in an approximation error that is large a priori, as shown in Nguyen et al. (2020), posterior inference is usually robust and no significant difference is detected. The choice of fixing K also allows for (much) faster code since the number of parameters is now fixed, and our implementation can thus take advantage of modern parallelization and vectorization algorithms. This is in line with our ultimate goal of fitting very large datasets with our model. In Appendix 8.C we also describe a slice sampling algorithm based on Griffin and Walker (2011) that does not require truncating the random measure.

8.3.1 MCMC Algorithm for the Truncated Model

Observe that in (8.8), $\theta_{ji} = \theta^*_k$ with positive probability. Therefore an alternative representation is achieved by introducing latent cluster indicator variables c_{ji} such that c_{ji} are independent categorical variables with support $\{1, \ldots, K\}$ and

$$P(c_{ji} = k \mid \{\lambda_{jh}\}, \{m_{hk}\}, \{J_k\}) \propto (\Lambda M)_{jk} J_k.$$

Let $T_j := \sum_k (\Lambda M)_{jk} J_k$. Writing $p(\cdot \mid \cdot)$ for a generic conditional density, the joint distribution of data and parameters under (8.8) is then

$$p(\{y_{j,i}\}, \{c_{j,i}\}, \{\lambda_{j,h}\}, \{m_{h,k}\}, \{J_\ell\}, \{\theta^*_\ell\}) = \\ \prod_{j=1}^{g} T_j^{-n_j} \prod_{i=1}^{n_j} f(y_{j,i} \mid \theta^*_{c_{j,i}}) (\Lambda M)_{j,c_{j,i}} J_{c_{j,i}} \times \prod_{h=1}^{K} \left[G_0(\theta^*_h) p_J(J_k) \prod_{k=1}^{K} \mathrm{Ga}(m_{hk} \mid \phi) \right] \pi(\Lambda)$$

To facilitate posterior inference, we introduce a set of auxiliary variables u_j, which are gamma distributed with shape parameter T_j and rate parameter n_j. Then

$$p(\{y_{j,i}\}, \{c_{j,i}\}, \{\lambda_{j,h}\}, \{m_{h,k}\}, \{J_\ell\}, \{\theta^*_\ell\}, \{u_j\}) = \\ \prod_{j=1}^{g} \frac{1}{\Gamma(n_j)} u_j^{n_j - 1} \prod_{i=1}^{n_j} f(y_{j,i} \mid \theta^*_{c_{j,i}}) (\Lambda M)_{j,c_{j,i}} J_{c_{j,i}} \times \exp\left(-\sum_{j=1}^{g} u_j \sum_{\ell=1}^{K} (\Lambda M)_{j,\ell} J_\ell \right) \\ \prod_{h=1}^{K} \left[G_0(\theta^*_h) p_J(J_k) \prod_{k=1}^{K} \mathrm{Ga}(m_{hk} \mid \phi) \right] \pi(\Lambda)$$

It is then possible to sample from the posterior distribution via a Gibbs sampler:

1. Update the atoms from

$$p(\theta^*_h \mid \cdots) \propto \prod_{j=1}^{g} \prod_{i : c_{j,i} = h} f(y_{j,i} \mid \theta^*_h) G_0(\theta^*_h)$$

2. Update the J's from

$$p(J_\ell \mid \cdots) \propto J_\ell^{q_\ell} \exp\left(-\sum_{j=1}^{g} u_j (\Lambda M)_{j,\ell} J_\ell \right) p_J(J_\ell)$$

   where $q_\ell = \sum_{j=1}^{g} \sum_{i=1}^{n_j} I[c_{j,i} = h]$.

3. Update the m's from

$$p(M \mid \cdots) \propto \prod_{j=1}^{g} \prod_{\ell=1}^{K} (\Lambda M)_{j,\ell}^{q_\ell} \times \exp\left(-\sum_{j=1}^{g} u_j (\Lambda M)_{j,\ell} J_\ell \right) \times \prod_{h=1}^{H} \prod_{k=1}^{K} \mathrm{Ga}(m_{hk} \mid \phi)$$

   The update of M can be done in a single block via Hamiltonian Monte Carlo.

4. Update the λ's from

$$p(\Lambda \mid \cdots) \propto \prod_{j=1}^{g} \prod_{\ell=1}^{K} (\Lambda M)_{j,\ell}^{q_\ell} \times \exp\left(-\sum_{j=1}^{g} u_j (\Lambda M)_{j,\ell} J_\ell \right) \pi(\Lambda)$$

   Again, we can update Λ using a single step of Hamiltonian Monte Carlo.

5. Update the cluster indicators from a categorical distribution over $\{1, \ldots, K\}$ with weights

$$P(c_{j,i} = h \mid \cdots) \propto f(y_{j,i} \mid \theta^*_h) (\Lambda M)_{j,h} J_h$$

6. update the u's from $u_j \mid \cdots \sim \mathrm{Gamma}(n_j, T_j)$

Finally, when the prior for Λ is the multiplicative gamma process (8.3) we propose to gain computational efficiency by selecting H through an adaptive Gibbs sampling scheme as in Bhattacharya and Dunson (2011). In particular, when adaptation occurs, we look at the "empty columns" of Λ. We define a column h of Λ to be empty if

$$\sum_{j=1}^{g} \frac{\lambda_{jh}}{\sum_{k=1}^{H} \lambda_{jk}} < \varepsilon \bar{\lambda}$$

where $\bar{\lambda} = H^{-1} \sum_{h=1}^{H} \sum_{j=1}^{g} \frac{\lambda_{jh}}{\sum_{k=1}^{H} \lambda_{jk}}$. In our experience $\varepsilon = 0.05$ provides satisfactory results. If there are no empty columns, we add a column sampled from the prior to Λ and a row sampled from the prior to M. Instead, if empty columns are found, we drop them from Λ and the corresponding rows from M.

Bhattacharya and Dunson (2011) propose to adapt Λ at each iteration ℓ with a probability p_ℓ that decreases exponentially fast. This choice is possible also within our algorithm but, in our experience, it significantly impacts run-time. This is due to the choice of using HMC to sample Λ and M and, in particular, to the use of the `tensorflow-probability` Python package, in combination with `LAX` compilation. For technical reasons, every time the size of Λ and M change, big chunks of the code must be recompiled, so that it's not efficient to adapt every few iterations. Instead, we propose to have a fixed adaptation window of $1,000$ iterations, where the adaptation occurs every 50 iterations. In our experience, this simple modification reduces the overall runtime by at least one order of magnitude.

8.4 Resolving the non-identifiability via post-processing

As already mentioned in the introduction, our model is not identifiable due to the multiplicative relation between Λ and $(\mu_1^*, \dots, \mu_h^*)$. This is not surprising, as the same holds for common latent factor models (Geweke and Singleton, 1980), where the likelihood is invariant to the action of orthogonal matrices. In that context, a common practice to recover identifiability is to constrain the matrix Λ to be lower triangular with positive entries on the diagonal (Geweke and Zhou, 2015). More recently, it has been proposed to ignore the identifiability issue and obtain a point-estimate of the posterior distribution either by post-processing the MCMC chains (see Papastamoulis and Ntzoufras, 2022; Poworoznek et al., 2021, and the references therein) or by choosing the maximum a posteriori (Schiavon et al., 2022). In particular, Poworoznek et al. (2021) propose to orthogonalize each posterior sample of Λ and then solve the sign ambiguity and label switching via a greedy matching algorithm.

The non-identifiability in our model is more severe than the one of common latent factor models. In fact, for any Q s.t. Q^{-1} is well defined, the likelihood is invariant when considering $\Lambda' = \Lambda Q^{-1}$ and $M' = QM$. Nonetheless, the constraints that $\Lambda' \geq 0$ (element-wise) and $M' \geq 0$ greatly reduce the number of matrices Q that can cause non-identifiability. In particular, we don't need to worry about sign ambiguity.

8.4.1 The Objective Function

Consider equation (8.2). Factorizations of the kind $\Gamma = \Lambda M$ where all the three matrices have nonnegative entries are common in blind source separation (BSS) problems, where the goal is to estimate "source components" M and "mixing proportions" Λ such that the observed signal Γ is approximately ΛM. Two well-established approaches to BSS are nonnegative matrix factorization (NMF, Sra and Dhillon, 2005) and independent component analysis (ICA, Hyvärinen, 2013). The main difference between the two consists in the loss function optimized. In NMF it is usually the norm of the approximation error, while, in ICA, the mutual information between the source components is minimized alongside the approximation error. This takes into account the goal of separating the components. Since in our analogy the sample size of the latent factor model is just one (i.e., in our model there is one single vector $\widetilde{\mu}_1, \dots, \widetilde{\mu}_p$ instead of multiple realizations), it is not possible to use the same criteria of ICA to define what we mean by "separated components". Hence, we propose to optimize with respect to the following *interpretability* criterion:

$$L(Q; M, J, \theta) = \sum_{i<j} \left(\int_{\mathbb{Y}} \left[\int_{\Theta} f(y \mid \theta) \mu'_i(\mathrm{d}\theta) \right] \left[\int_{\Theta} f(y \mid \theta) \mu'_j(\mathrm{d}\theta) \right] \mathrm{d}y \right)^2 . \tag{8.9}$$

where

$$\mu'_j = \sum_{k=1}^{K} (QM)_{jk} J_k \delta_{\theta^*_k}$$

Intuitively, low values of $L(Q; M, J, \theta)$ in (8.9) are attained when the transformed random measures μ'_h, mixed with the mixture kernel f, result in well separated densities.

Defining $g_i(y) := \int_\Theta f(y \mid \theta)\mu'_i(\mathrm{d}\theta)$ it is clear that (8.9) can be interpreted as the sum of the squared inner products (in the L_2 sense) between g_i and g_j. The L_2 distance is not commonly used to measure the discrepancy of densities. A more familiar option would be to consider $\int \sqrt{g_i(y)}\sqrt{g_j(y)}\mathrm{d}y$, that is $1 - d_{\mathcal{H}}(g_i, g_j)$ where $d_{\mathcal{H}}$ denotes the Hellinger distance. However, this choice of loss function leads to a more complex optimization problem, that cannot be solved with our approach. Indeed, as discussed later in Section 8.4.3, the positivity of the density g_i might not be preserved by the intermediate steps of the algorithm. Therefore, we need a loss function that continues to make sense for negative densities.

8.4.2 The Optimization Space

Consider now the space over which one should minimize (8.9). First of all, we must require the existence of Q^{-1} to interpet $\Lambda' = \Lambda Q^{-1}$. Moreover, for the model to make sense we need to ensure the positivity of the coefficients involved, i.e. $\Lambda' = \Lambda Q^{-1} \geq 0$ and $M' = QM \geq 0$. Finally, we observe that (i) given an "optimal" Q such that $L(Q; M, J, \theta) = 0$, $L(\gamma Q; M, J, \theta) = 0$ for any $\gamma > 0$, and (ii) $L(Q; M, J, \theta)$ attains lower values when the entries in Q are small. Despite the preference for small Q in the optimization problem, the resulting model is invariant to such rescalings since it involves the normalization of the underlying random measures. Hence, to overcome both issues we propose to add a further constraint in the optimization problem, namely $\det Q = 1$, which prevents having several optimal solutions differing by a constant and does not allow for matrices with entries too close to 0.

In conclusion, we propose to optimize (8.9) over the special linear group $SL(H) = \{Q \in \mathbb{R}^{H \times H} : \det Q = 1\}$, with the additional positivity constraints, i.e. our optimization problem becomes

$$\min_{Q \in SL(H)} \sum_{h,k=1}^{H} L(Q; M, J, \theta) \quad \text{s.t.} \quad \Lambda Q^{-1} \geq 0, \ QM \geq 0. \tag{8.10}$$

The special linear group is not a linear space, therefore common gradient-based optimization techniques cannot be used to solve (8.10). However, we can take advantage of the differential structure of $SL(H)$. In fact, it is a Lie group (hence, a smooth differentiable Manifold) with associated Lie algebra $\mathfrak{sl}(H) = \{A \in \mathbb{R}^{H \times H} : \mathrm{tr} A = 0\}$. See Appendix 8.A.2 for some basic details regarding Riemannian manifolds and Lie groups.

Algorithm 1. Augmented Lagrangian Multiplier Method

```
input Starting point $Q$, initial values $\rho$, $\gamma_j$, target threshold $\varepsilon^*$, initial threshold $\varepsilon$.
repeat
    $Q = Q'$
    solve $Q' = \arg\min_Q \mathcal{L}_\rho(Q, \gamma)$ for fixed $\rho, \gamma$ with theshold $\varepsilon$ using Algorithm 2
    $\gamma_j = \gamma_j + \rho c_j(Q')$
    $\rho = 0.9\rho$ $\varepsilon = \max\{\varepsilon^*, 0.9\varepsilon\}$
until $\varepsilon \leq \varepsilon^*$; $\|Q - Q'\| \leq \varepsilon$
end
```

Algorithm 2. Lie RATTLE Optimization

```
input Starting point $Q, P$, momentum $\tau$, stepsize $s$, threshold $\varepsilon$.
repeat
    $P = \tau\left(P - s\Pi_{\mathfrak{sl}(H)}(\partial_Q \mathcal{L}_\rho(Q, \gamma), Q)\right)$
    $Q = Q\exp_m(\chi P)$, $\chi = \cosh(-\log\tau)$
    $P = \tau\left(P - s\Pi_{\mathfrak{sl}(H)}(\partial_Q \mathcal{L}_\rho(Q, \gamma), Q)\right)$
until $\|Q - Q'\| \leq \varepsilon$
end
```

8.4.3 A Riemannian Augmented Lagrangian Method

We are now in place to state the algorithm. For notational convenience, define the functions $c^1_{jh}(Q) = -(\Lambda Q^{-1})_{jh}$ and $c^2_{hk} = -(QM)_{hk}$. Denote with c_j the collection of all such functions. The positivity constraints are equivalent to $c_j \leq 0$ for all j's. Following the augmented Lagrangian method (Birgin and Martinez, 2014), we can deal with the constraints $\Lambda Q^{-1} \geq 0$ and $QM \geq 0$ by introducing auxiliary parameters ρ, γ_j and define the augmented loss function

$$\mathcal{L}_\rho(Q, \gamma) = L(Q; M, J, \theta) + \frac{\rho}{2}\sum_j \max\left\{0, \frac{\gamma_j}{\rho}c_j(Q)\right\} \tag{8.11}$$

Then, we can solve (8.10) by alternating between minimizing (8.11) for fixed values of ρ, γ_j and updating ρ, γ_j as in Algorithm 1. As in the usual augmented Lagrangian method, the constraints might be violated in the intermediate steps. Intuitively, the fact that the penalty term γ_j is increased at every iteration if the constraint is violated should force the solution of the problem inside the feasible region. See Birgin and Martinez (2014) for convergence results of the augmented Lagrangian method.

It is now left to discuss how to solve (8.11) for fixed ρ and γ_j. We propose to tackle this problem with the Riemannian dissipative RATTLE algorithm in França et al. (2021), reported for the special case of optimization over $SL(H)$ in Algorithm 2. In particular, $\Pi_{\mathfrak{sl}(H)}$ is the projection over the Lie algebra $\mathfrak{sl}(H)$ while $\exp_m$ denotes the matrix exponential, which is a map $\mathfrak{sl}(H) \to SL(H)$. Informally, Algorithm 2 resembles an accelerated gradient method, where a momentum term is introduced to speed up the convergence. We further have

$$\partial_Q \mathcal{L}_\rho(Q, \gamma)_{ij} = \frac{\partial_Q \mathcal{L}_\rho(Q, \gamma)}{\partial Q_{ji}}$$

(note the index flip $ij \to ji$, in other words $\partial_Q f(Q) = \nabla_Q f(Q)^\top$ where ∇ stands for the usual Euclidean gradient). Moreover, the following proposition gives a computationally convenient way of evaluating $\Pi_{\mathfrak{sl}(H)}$.

Proposition 8.4. *Let X an $H \times H$ real valued matrix. Then*

$$\Pi_{\mathfrak{sl}(H)}(X) = (X - diag(X))^T + \sum_{\ell=1}^{H-1} X^*_\ell$$

where $diag(X)$ is the diagonal matrix with entries equal to the diagonal of X and X^*_ℓ is a diagonal matrix whose only nonzero entries are the (ℓ, ℓ)-th and the $(\ell+1, \ell+1)$-th ones, which equal to $X_{i,i} - X_{i+1,i+1}$ and $-X_{i,i} - X_{i+1,i+1}$ respectively.

The parameters involved in the optimization problem are: the stepsize s and momentum factor τ in Algorithm 2 as well as the initial values ρ, γ_j and the target and thresholds ε^*, ε in Algorithm 1. We suggest as defaults $s = 10^{-6}$, $\tau = 0.9$, $\rho = \gamma_j = 10$, $\varepsilon^* = 10^{-6}$, $\varepsilon = 10^{-2}$. Finally, to set the starting point Q we we solve the unconstrained optimization problem (equivalent to setting $\gamma_j = 0$ in (8.11)) using Algorithm 2 and use that solution as starting point for the constrained optimization. The initial momentum term P in Algorithm 2 is always the zero matrix.

8.4.4 The Label-Switching Problem

Observe that another source of non-identifiability comes from the labeling of $\mu^*_1, \ldots, \mu^*_H$. Namely, the likelihood and the loss function (8.9) are invariant under permutation of the indices$\{1, \ldots, H\}$, provided that the columns of Λ are permuted as well. This prevents the possibility of computing reliable posterior summaries of the μ^*_h's and Λ from the MCMC chains.

We propose to post-process the output of our sampling algorithm to get rid of this problem. In particular, as in Poworoznek et al. (2021), we propose to align the latent measures at each iteration to a given template. Let $\hat{\mu}_1, \ldots, \hat{\mu}_H$ denote the template. For instance,

$$\hat{\mu}_h = \sum_{k=1}^{K} (Q^{(\ell)} M^{(\ell)})_{jk} J^{(\ell)}_k \delta_{\theta^{(\ell)}_k}$$

where we denote with the superscript ℓ the index of the MCMC sample. We choose ℓ to approximate the maximum a posteriori. $Q^{(\ell)}$ denotes the associated optimal transformation matrix obtained as outlined above. Let $d(\hat{\mu}_h, \mu'_j)$ denote a dissimilarity between two measures. Two specific choices are discussed later. We align each $(\mu'^{(j)}_1, \ldots, \mu'^{(j)}_H) := Q^{(j)}(\mu^{*(j)}_1, \ldots, \mu^{*(j)}_H)$ to $\hat{\mu}_1, \ldots, \hat{\mu}_H$ by learning an optimal permutation σ of $\{1, \ldots, H\}$, associated to a permutation matrix P_σ that minimizes $\sum_h d(\hat{\mu}_h, \mu^{(j)\prime}_{\sigma(h)})$ by solving

$$\inf_{P \in \text{Perm}_H} \sum_{h,k=1}^{H} d(\hat{\mu}_h, \mu^{(j)\prime}_k) P_{hk}$$

where Perm_H denotes the space of $H \times H$ permutation matrices. Naively, this would require $H!$ computations. Instead, we solve the relaxed optimization problem by looking for the P stochastic matrix (i.e., rows and columns sum to one) that minimizes the objective above. That is, we solve for the Wasserstein distance between the empirical measures ν_1 and ν_2 defined as

$$\nu_1 = \frac{1}{H} \sum_{h=1}^{H} \delta_{\hat{\mu}_h}, \qquad \nu_2 = \frac{1}{H} \sum_{k=1}^{H} \delta_{\mu^{(j)\prime}_k}$$

where ν_i is a probability measure on the space of positive measures over Θ. Birkhoff's theorem ensures that the solution to the relaxed optimization problem is a permutation matrix.

As far as the dissimilarity $d(\hat{\mu}_h, \mu'_j)$ is concerned, in our examples we considered

$$d(\hat{\mu}_h, \mu'_j) = \left\| \hat{\mu}_h(\Theta)^{-1} \int_\Theta f(y \mid \theta)\hat{\mu}_h(\mathrm{d}\theta) - \mu'_j(\Theta)^{-1} \int_\Theta f(y \mid \theta)\mu'_j(\mathrm{d}\theta) \right\|$$

where $\| \cdot \|$ stands for the L_2 norm. This distance requires the numerical evaluation of a mixture density on a fixed grid, to compute the associated L_2 distance. This is easy when the dimension of the data space is small, typically when data are uni or bi-dimensional. See Appendix 8.D for a more efficient alternative in higher dimensions.

8.5 Simulation Study

We present two simulations to assess the performance of our model. In all the examples, we consider Gaussian mixture models, i.e., $\theta^*_h = (\mu_h, \sigma^2_h)$ and $f(\cdot \mid \theta) = \mathcal{N}(\cdot \mid \mu, \sigma^2)$. The scores m_{hk} in the CoRM are gamma distributed and each μ^*_h is marginally a gamma process (before the truncation) with total mass equal to 1 and base measure equal to the Normal-inverse-Gamma distribution, i.e. $G_0(\mu, \sigma^2) = \mathcal{N}(\mu \mid \mu_0, \sigma^2/\lambda) IG(\sigma^2 \mid a, b)$. We set μ_0 equal to the empirical mean of the observations, $\lambda = 0.01$, $a = b = 2$. We truncate the CoRM to $K = 20$ jumps to perform posterior inference. Specific choices of the prior for Λ are discussed case-by-case.

8.5.1 Interpretation of the posterior distribution

Before giving details on the numerical illustration, we discuss how to obtain interpretable summaries of the posterior distribution, after post-processing. This also allows us to set some notation used in the next sections.

Interpreting the unnormalized *latent factor densities* $\int_\Theta f(\cdot \mid \theta)\mu^*_h(\mathrm{d}\theta)$ is difficult because of the lack of a common scale to which the densities should be referred. In fact, note that these are not probability densities. Let p_j be the j-th group-specific density. We can write

$$p_j = \int_\Theta f(\cdot \mid \theta)\bar{\tilde{p}}(\mathrm{d}\theta) + \sum_{h=1}^{H} s_{jh} \int_\Theta f(\cdot \mid \theta)\epsilon_h(\mathrm{d}\theta)$$

where $\bar{\tilde{p}}(\mathrm{d}\theta)$ is the average of $\tilde{p}_1, \ldots, \tilde{p}_g$, $p'_h = \mu'_h/\mu'_h(\Theta)$, $\epsilon_h = p'_h - \bar{\tilde{p}}(\mathrm{d}\theta)$ and the scores s_{jh}'s are defined as

$$s_{jh} = \frac{\lambda'_{jh}\mu'_h(\Theta)}{\sum_{k=1}^{H} \lambda'_{jk}\mu'_k(\Theta)} \tag{8.12}$$

Note that ϵ_h is a signed measure. Instead of comparing the latent factor densities, we find it considering the *residual factor densities* $\int_\Theta f(\cdot \mid \theta)\epsilon_h(\mathrm{d}\theta)$ leads to easier interpretations.

Moreover, we can associated to each μ'_h an *importance score* I_h defined as $I_h = \sum_{j=1}^{g} s_{jh}$ The rationale comes from writing $\mu'_h = \mu'_h(\Theta)p'_h$ so that

$$p_j = \int_\Theta f(\cdot \mid \theta) \sum_{h=1}^{H} \frac{\lambda'_{jh}\mu'_h(\Theta)}{\sum_{k=1}^{H} \lambda_{jk}\mu'_k(\Theta)} p'_h(\mathrm{d}\theta) = \sum_{h=1}^{H} s_{jh} \int_\Theta f(\cdot \mid \theta)p'_h(\mathrm{d}\theta)$$

that is, we express each $\widetilde{p}_j$ as a convex combination of probability measures and with weight s_{jh}.

With an abuse of notation, we will denote by μ'_h the posterior mean of $(\mu^*_1, \ldots, \mu^*_h)$ and with Λ' the posterior mean of Λ, obtained after the post-processing of the MCMC chains, that is

$$\mu'_h = \frac{1}{M} \sum_{\ell=1}^{M} \sum_{k \geq 1} \left(P^{(\ell)} Q^{(\ell)} M^{(\ell)}\right)_{hk} J^{(\ell)}_k \delta_{\theta^{*(\ell)}_k}, \quad \Lambda' = \frac{1}{M} \sum_{\ell=1}^{M} \left(\Lambda^{(\ell)} (Q^{(\ell)})^{-1}\right) (P^{(\ell)})^\top \tag{8.13}$$

where the superscript ℓ, $\ell = 1, \dots, M$ is used to denote the iteration of the MCMC algorithm, $Q^{(\ell)}$ is the matrix found with Algorithm 1, and $P^{(\ell)}$ is the permutation matrix found as in Section 8.4.4.

8.5.2 Only Group Information

We consider here a simulated example with $g = 100$ groups of data, where each $n_j = 25$. We consider the situation where we tend to observe only small differences across populations by considering the following data generation process

$$y_{j,i} \overset{\text{iid}}{\sim} w_{j1}\mathcal{N}(-2,2) + w_{j2}\mathcal{N}(0,2) + w_{j1}\mathcal{N}(2,2), \qquad i = 1, \dots, n_j$$

and for each group we simulate $\boldsymbol{w}_j = (w_{j1}, w_{j2}, w_{j3}) \overset{\text{iid}}{\sim} \text{Dirichlet}(1,1,1)$. In most of the groups, the data generating density is unimodal and they differ mainly because of different levels of skewness.

As prior for Λ, we assume the multiplicative gamma process (8.3) setting $H = 20$. We run the MCMC chains for a total of $11,000$ of which the first $1,000$ are used for the adaptation and the following $5,000$ are discarded as burn-in. The adaptation phase quickly finds between 3 and 5 latent measures, 4 being the final value. We post-process the chains as in Section 8.4.

Figure 8.5.1 shows the inferred latent factors before and after the post-processing. It is clear that solving the label switching is essential. Although not particularly evident from the plot, the matrices $Q^{(j)}$ found by the optimization algorithm were significantly different from the identity, hence showing the usefulness of the post-processing. Our approach identifies the main common traits in the data. Factors 1 and 3 peak around -2 and 2 respectively, while the second and fourth factors are both more concentrated around the origin, with the second one presenting a light skewness and heavier right tail. The residual factor densities can be used to infer the same description of the latent measures.

8.5.3 Area-Referenced Data

We consider data over a regular lattice on $0, 1, \dots, q \times 0, 1, \dots, q \subset \mathbb{Z}^2$. We consider $q = 4, 8, 16$ so that the number of groups is $g = 16, 64, 256$ respectively. Following the simulation study in Beraha et al. (2021), we generate data at each location from a three-component Gaussian mixture with means $-5, 0, 5$ respectively and variances equal to one. Let x_j, y_j denote the x and y coordinate of location j on the lattice. The location-specific weights are

$$(w_{j1}, w_{j2}, w_{j3}) = \left(e^{\widetilde{w}_{j1}}, e^{\widetilde{w}_{j2}}, 1\right) / \left(1 + e^{\widetilde{w}_{j1}} + e^{\widetilde{w}_{j2}}\right)$$

where

$$\widetilde{w}_{j1} = 3(x_j - \bar{x}) + 3(y_j - \bar{y}), \quad \widetilde{w}_{j2} = -3(x_j - \bar{x}) - 3(y_j - \bar{y})$$

and $(\bar{x}, \bar{y})$ denote the center of the lattice. For each location, 25 observations are simulated.

We compare our model with prior (8.4) for $H = 1, 2, 3, 5, 10$ with the spatially dependent mixture model (SPMIX, Beraha et al., 2021) and the Hierarchical Dirichlet Process (HDP, Teh et al., 2006). Although the latter does not take into account the spatial dependence, it is shown in Beraha et al. (2021) that the HDP performs well when the number of groups g is small.

We truncate the CoRM to $K = 20$ jumps and set the number of components in SPMIX to 20 as well. Prior distributions can be assumed for τ and ρ in (8.4). However, since the likelihood is invariant with respect to rescalings of Λ, we found that having a prior on τ led to non-convergent MCMC chains for Λ. In particular, after a few thousand iterations, the values of the entries in Λ were in the order of 10^{100}. Hence, we suggest

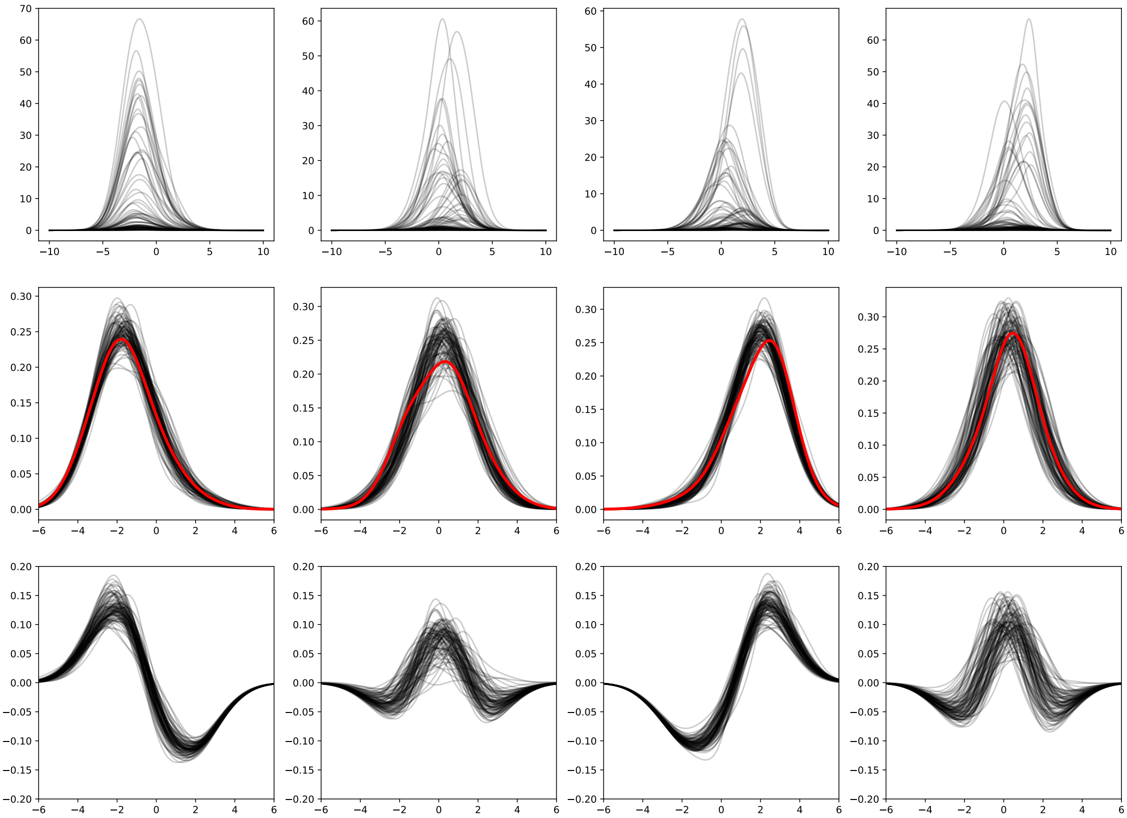


Figure 8.5.1: Posterior summaries for the simulation in Section 8.5.2. Top row: draws from the posterior distribution of the latent factor densities. Middle row: draws after post-processing and normalization, the red density denotes the template. Bottom row: posterior draws of the residual factor densities.

fixing τ so a sufficiently large value. In our simulations, we always set $\tau \equiv 2.5$. Assuming a prior for ρ does not have such an impact on posterior inference. However, it would require re-computing the determinant of Σ^{-1} at every MCMC iteration, which requires $O(g^3)$ operations. Hence, we fix ρ to 0.95 to encourage strong spatial dependence in our examples. Another possibility would be to fix a grid of values in $(0, 1)$ and assume a discrete prior for rho over it, allowing to compute all the required matrix determinants beforehand.

All the MCMC chains are run for $10,000$ iterations, discarding the first $5,000$ as burn-in. It is clear from Figure 8.5.2 (top row) that our model outperforms the competitors when $g = 16, 64$ and performs slightly better than the spatial mixture model when $g = 256$. In all the settings, the best performance is associated with $H = 3$ latent measures. Posterior samples of the latent factor densities are reported in Figure 8.5.2 (bottom row) for the setting with $g = 64$ and $H = 3$. In this case, the latent densities are already well separated so that there is no need to post-process the MCMC chains using the algorithm described in Section 8.4. The three latent densities give mass to one of the three modes in the data each.

8.6 Real Data Illustrations

In this section, we illustrate our methodology on two real datasets. In both cases, data are univariate and we let $f(\cdot \mid \theta)$ be the Gaussian density with parameters $\theta = (\mu, \sigma^2)$. The

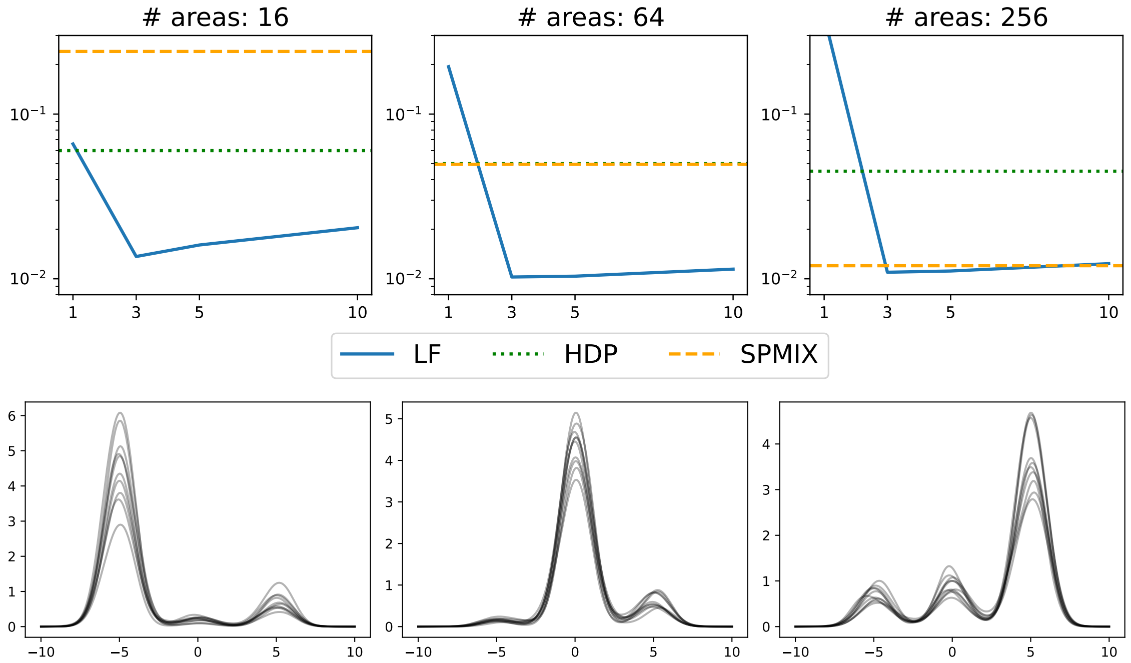


Figure 8.5.2: Top row: Average Kullback–Leibler divergence between the true data generating density and the Bayesian estimate, as a function of the number of latent measures H. From left to right $g = 16, 64, 256$. Bottom row: Posterior samples for the latent factor densities when $g = 64$ and $H = 3$

base measure G_0 is the Normal-inverse-Gamma distribution, whose parameters are set as in Section 8.5. Moreover, we always truncate to $K = 20$ points the support of the random measures.

8.6.1 The Invalsi Dataset

We consider the *Invalsi* dataset[1] that collects the evaluation of a unified math test undertaken by all Italian high-school students. Grades vary from 1 to 10 with 6 being the passing grade. We pre-process the data by adding a small Gaussian noise with zero mean and standard deviation equal to 0.25. The dataset contains the scores of 39377 students, subdivided into 1048 schools. The number of students per school varies from 4 to 131, with 37 students per school on average with a standard deviation of 12 approximately.

We assume the multiplicative gamma process prior for Λ as in (8.4) with $H = 20$. The initial adaptation phase identifies 5 latent factors. Draws from the latent factor densities are displayed in Figure 8.6.1. It is clear that some label switching is happening between the fourth and fifth factors. After the post-processing, for ease of visualization, we discretized the estimated normalized latent factor densities to the original grades $i = 1, \ldots, 10$ by evaluating $\int_{i-0.5}^{i+0.5} f(y \mid \theta)\mu'_h(\mathrm{d}\theta)/\mu'_h(\Theta)$. The estimated factors are displayed in the first two rows of Figure 8.6.1. They represent a wide range of behaviors: the first one is concentrated on negative grades below the passing threshold, the second one is centered on the passing grade, and the third one on grades way above the passing grade. The fourth and the fifth represent more complex distributions: the former one covering the range of "just below the passing grade and just above it", the latter one instead represents a distribution peaked at 5 with a heavy right tail.

[1] available for research purposes at https://invalsi-serviziostatistico.cineca.it

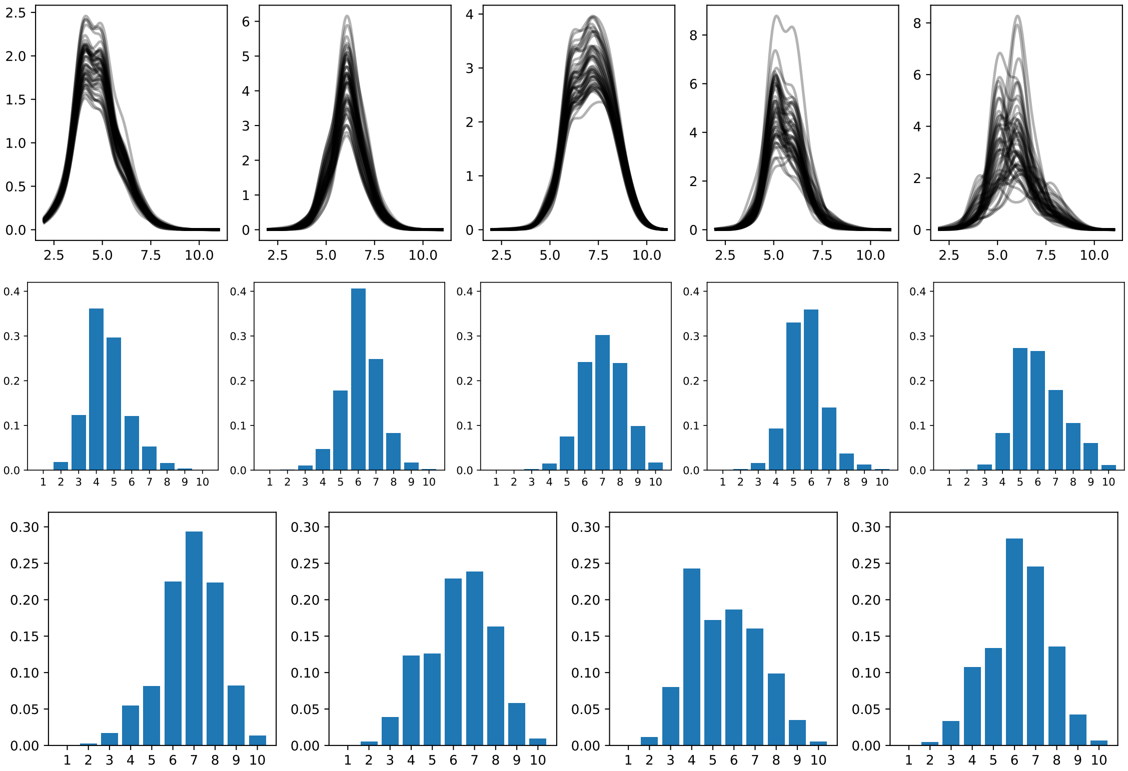


Figure 8.6.1: Summary of posterior inference on the Invalsi dataset. Top row: draws from the posterior distribution of the latent factor densities. Middle row: estimates of the discretized normalized latent factor densities after post-processing. Bottom row: average density in each cluster discredized on the intervals $[i-0.5, i+0.5)$, $i = 1, \dots, 10$.

The importance scores I_h are approximately $331, 184, 351, 165, 16$. Hence, we can interpret that the two most relevant common traits are the ones represented by μ'_1 (that combines a sharp peak in 4, with a heavy right tail), and by μ'_3, which gives mass to grades above the passing threshold.

Finally, we look at the scores λ_{jh}'s after the post-processing. We can understand the similarities between schools by clustering the scores for each school from the corresponding row of the matrix Λ'. Using a hierarchical clustering algorithm yields four clusters (the dendrogram is shown in Figure 8.E.4 in the Appendix). We then compute the average value $\hat{\lambda}_\ell = (\hat{\lambda}_{\ell 1}, \dots, \hat{\lambda}_{\ell H})$ for each of the four clusters, to which a probability measure $\tilde{p}_\ell \propto \sum_{h=1}^{H} \hat{\lambda}_{\ell h} \mu'_h$ and report the associated mixture density in the bottom row Figure 8.6.1.We define a cluster-specific mean distributions $\tilde{p}_\ell \propto \sum_{h=1}^{H} \hat{\lambda}_{\ell h} \mu'_h$ by taking the average value $\hat{\lambda}_\ell = (\hat{\lambda}_{\ell 1}, \dots, \hat{\lambda}_{\ell H})$ for each of the four cluster. the associated mixture densities are shown in the bottom row Figure 8.6.1. The clusters are easily interpretable and the mean distributions $\tilde{p}_1, \dots \tilde{p}_4$ are substantially different.

8.6.2 Californian Income Data

We consider the 2021 ACS census data publicly available at https://www.census.gov/programs-surveys/acs/data/experimental-data/2020-1-year-pums.html. Specifically, we consider the `PINCP` variable that represents the personal income of the survey responders and restrict to the citizens of the state of California. For privacy reasons, data are grouped into geographical units denoted as *PUMAs*, roughly corresponding to $100,000$

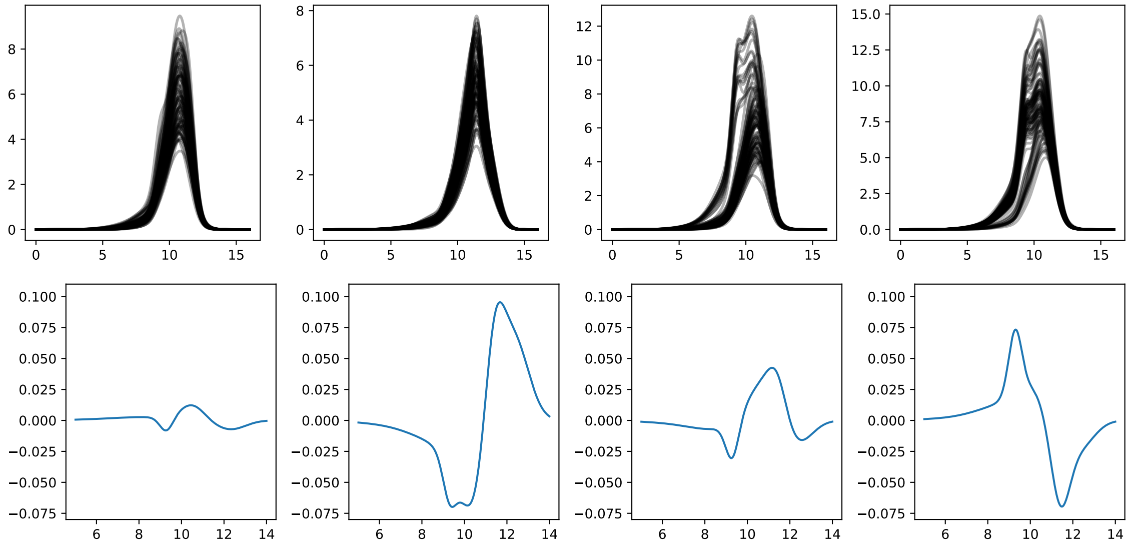


Figure 8.6.2: Summary of posterior inference on the Californian income dataset. Top row: draws from the posterior distribution of the latent factor densities. Bottom row: average of the residual factor densities after post-processing.

inhabitants. There are 265 PUMAs in California. We consider $y_{j,i}$ to be the logarithm of the income of the i-th person in the j-th PUMA. The total number of responders is 43380, with the median number of observations per PUMA being 164.

As shown in Figure 8.E.5 in the Appendix, the distributions of the income in different PUMAs are quite varied with clear spatial dependence. This is also confirmed by the analysis of Moran's I index for the average log-incomes, which is approximately 0.55. A permutation test confirmed that the spatial correlation is not-negligible. We assume independent log Gaussian Markov random fields priors for each column of Λ as in (8.4), where we fix $\tau = 2.5$ and $\rho = 0.95$. We choose H by evaluating the predictive goodness of fit for $H = 1, \dots, 10$ using the widely applicable information criterion (WAIC, Watanabe, 2013). The best performance is associated with $H = 4$, therefore we comment on the posterior inference obtained under this model.

Figures 8.6.2 and 8.6.3 summarize the posterior findings. The draws from the latent measures (top row) show some evidence of label-switching in the third and fourth factors. Post-processing the chains with our algorithm estimates the four latent factors in Figure 8.E.6 in the Appendix. However, it is easier to interpret the residual factor densities displayed in the bottom row of Figure 8.6.2. The second and the fourth factors are associated with the largest variations. In particular, the second one gives mass to higher incomes while the fourth one gives mass to lower incomes. The first one is more representative of the average population since the variations are small. The third factor instead corresponds to average incomes and gives less mass (compared to the average population) to both low and high incomes. To visualize the spatial effect of the latent factors, we plot the scores s_{jh} for each factor. Note that the third latent factor is predominant in several areas, where s_{j3} is larger than 0.8. Instead, s_{j2} is small in all of California except for a few PUMAs in San Francisco, Long Beach, and San Diego, where the highest incomes are observed. In particular, zooming on San Francisco (middle row of Figure 8.6.3), we note that the second factor is highly represented in Palo Alto, home to several tech tycoons, and San Rafael, home to entertainers. Finally, note that the fourth factor (associated with the lowest incomes) has a high weight in the two PUMAs neighboring Mexico as well in some areas in Los Angeles. Notably, the PUMA around the port and the one corresponding to

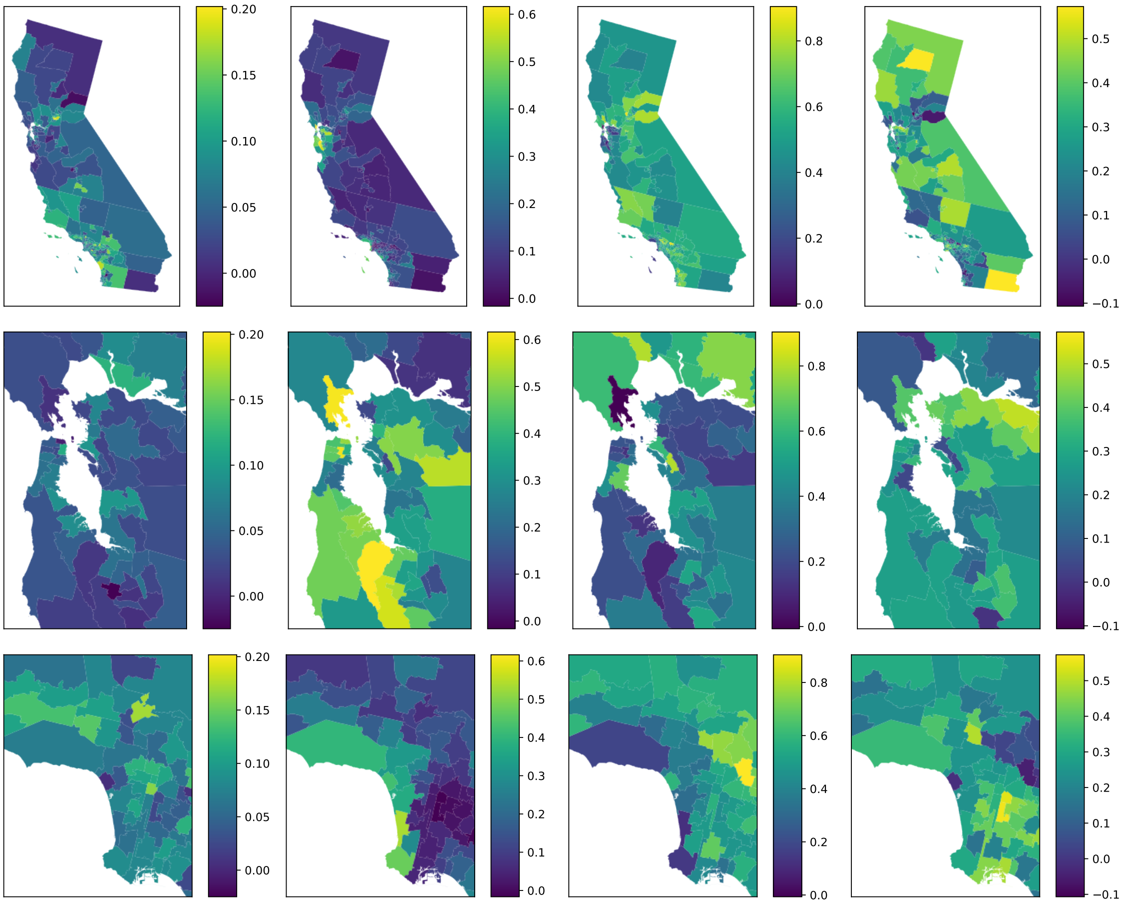


Figure 8.6.3: Spatial distribution of the scores in the Californian income dataset. Top row: the scores s_{jh} for $h = 1, \dots, 4$ from left to right. Middle row: zoom on the San Francisco area. Bottom row: zoom on the Los Angeles area

the "south LA" neighborhoods going from University Park to Green Meadows. This is in accordance with the 2008 *Concentrated Poverty in Los Angeles* report (Flaming and Matsunaga, 2008), which estimates that the percentage of households in poverty is typically above 40% in those areas.

8.7 Discussion

Modeling a collection of random probability measures is an old problem that has received considerable attention in the Bayesian nonparametric literature, see, e.g. Quintana et al. (2022) for a recent review. In this article, we have considered specifically the case when data are naturally divided into groups or subpopulations, and data are partially exchangeable. Taking a nonparametric Bayesian approach, we assumed that observations in each group can be suitably modeled by a mixture density, and proposed *normalized latent measure factor models* as a prior for the collection of mixing measures in each group. Similar to the Gaussian latent factor model, our model assumes that each group-specific directing measure is a linear combination of a set of latent random measures. We can interpret the latent random measures as the latent common traits shared by the subpopulations. Moreover, the prior for the linear combination weights can include additional group-specific information such as geographical location.

To account for the non-identifiability of our model, we developed an ad-hoc post-processing algorithm leading to a constrained optimization algorithm over the special linear group, that is the group of matrices whose determinant is equal to one. To solve the optimization problem, we leveraged recent work on optimization on manifolds, proposing a Riemannian augmented Lagrangian method. Through simulations and illustrations on two real datasets, we validate our approach and show its usefulness, focusing in particular on the interpretation of the latent measures and the associated weights. The model opens up many direction for future research which we discuss below and whic we aim to investigate thoroughly in the future.

The structure of our factor model approach allows it to be extended to a wide-range of dependence structures between the groups. For example, including observation-specific covariates in the model or time-dependent data. We can also build models which allow for the discovery of latent structure in the groups by further modelling the factor loadings matrix Λ. For instance, Rodriguez et al. (2008), Camerlenghi et al. (2019), and Beraha et al. (2021) build models which cluster groups according to the similarity of their distributions. We could this by assuming that each of the group-specific directing measures is equal to one of the latent measures, *i.e.* only one of $\lambda_{j1}, \ldots, \lambda_{jH}$ are non-zero, which would be similar to exploratory factor analysis (Conti et al., 2014). Alternatively, we can achieve a "soft clustering" of the group-specific distributions by assuming a mixture model for the rows of the matrix Λ. More generally, Λ could be expressed in terms of further low-rank matrix to find similarities between the group-specific factor loadings.

The post-processing identification scheme leads to estimated latent factor densities which are maximally separated according to the interpretability criterion. This allows us to interpret the factor loadings as an H-dimensional summary of the group-specific distribution where the each element of the summary measures different parts of the distribution. In a similar way to scores from dimension reduction techniques, such as Principal Components Analysis, or embeddings in machine learning, these estimates can then be used as inputs into other statistical analysis. We effectively use this idea in the analysis of the Invalsi data-set where the estimated factor loadings are clustered to find groups of schools with similar distributions. This approach could have much wider applications. For example, the analysis of the Californian income data leads to estimated factor loadings for each PUMA which could be used in a regression model in place of other summaries such as median income, or the percentage of incomes below/above a threshold. These estimated factor loadings should provide more information and a single measure and be a more efficient representation than a large number of measures (for example, using a large number of thresholds). It would be particularly interesting to investigate this approach for multivariate observations where it's difficult to find efficient low-dimensional summaries of distributions.

Appendix

8.A Technical Preliminaries

8.A.1 Completely Random Measures

Let $\mathbb{M}_\Theta$ be the space of boundedly finite (positive) measures over the space $(\Theta, \mathcal{B}(\Theta))$, where $\mathcal{B}(\Theta)$ is the Borel σ-algebra. We endow $\mathbb{M}_\Theta$ with the corresponding Borel σ-alebra $\mathcal{M}$. Then, a random measure is a measurable function from a base probability space $(\Omega, \mathcal{F}, \mathbb{P})$ to $(\mathbb{M}_\Theta, \mathcal{M})$.

Following Kingman (1967), we say that a random measure μ is completely random if, for any $\{A_1, \ldots, A_m\} \subset \mathcal{B}(\Theta)$, $A_i \cap A_j = \emptyset$ $(i \neq j)$, we have that the random variables $\mu(A_i)$, $i = 1, \ldots, n$ are independent.

For our purposes, it is sufficient to consider completely random measures of the kind

$$\mu(A) = \int_{\mathbb{R}_+ \times A} sN(\mathrm{d}s\mathrm{d}x)$$

where N is a Poisson point process on $\Theta \times \mathbb{R}_+$ with base (intensity) measure. We will assume that the intensity measure factorizes as $\nu(\mathrm{d}s)G_0(\mathrm{d}x)$ where ν is a Borel measure on the positive reals and G_0 is a probability measure on Θ. Then, the random measure $\mu(A)$ is uniquely characterized by its Laplace transform, for any measurable f, $f(x) \geq 0$:

$$\mathbb{E}\left[e^{-\int_\Theta f(x)\mu(\mathrm{d}x)}\right] = \exp\left(-\int_{\mathbb{R}_+ \times \Theta} \left(1 - e^{-sf(x)}\right) \nu(\mathrm{d}s)G_0(\mathrm{d}x)\right),$$

where the equality follows from the Lévy-Khintchine representation of the underlying Poisson process.

A key result that will be used later, is the Cambell-Little-Mecke formula (also referred to as the Palm formula) which allows the interchange of expectation and integral when the integrand measure is a point process. We report here the result for Poisson point processes, the most general case can be found in Baccelli et al. (2020).

Theorem 8.2. *[*Campbell-Little-Mecke*]*

Let N be a Poisson point process over a complete and separable metric space $\mathbb{X}$ with intensity measure $\nu(\mathrm{d}x)$. Denote by $\mathbb{M}_\mathbb{X}$ the space of boundedly σ-finite measures on $\mathbb{X}$. Then, for any measurable $g : \mathbb{X} \times \mathbb{M}_\mathbb{X} \to \mathbb{R}_+$ we have

$$\mathbb{E}\left[\int g(x, N)N(\mathrm{d}x)\right] = \int \mathbb{E}[g(xN + \delta_x)]\nu(\mathrm{d}x) \tag{8.14}$$

where both expectations are with respect to the law of the Poisson process N.

8.A.2 Riemannian Manifolds and Lie Groups

A group G is a set equipped with a binary operation: $G \times G \to G$ with the additional properties that the operation is associative, there exists an identity element and every element has its inverse. A Lie group arises if the set is a differentiable manifold and the binary and inverse operations are smooth differentiable functions. A classic example of a Lie group is the set of 2×2 real-valued invertible matrix, endowed with the group operation $(A, B) \mapsto AB$, that is the standard matrix multiplication. This group is usually referred to as the *general linear group* of dimension two and is denoted by $GL(2, \mathbb{R})$.

For our purposes, it is sufficient to consider matrix Lie groups, i.e., the case when G is a set of matrices, so that $G \subset \mathbb{R}^{n \times n}$ for some n. We can thus endow G with the Riemannian metric induced by the Euclidean metric in $\mathbb{R}^{n^2}$ Then G is a Riemannian manifold (it locally resembles a Euclidean space), and we can define at each point $g \in G$ a tangent space $T_g G$ together with the maps $\exp_g : T_g G \to G$ and $\log_g : G \to T_g G$.

The tangent spaces in Lie groups admit a particularly simple representation. Thanks to the fact that left multiplication by an element $g \in G$, that is the map $L_g(x) = gx$, is a diffeomorphism whose inverse is $(L_g)^{-1} = L_{g^{-1}}$, we have that the tangent space $T_g G$ at g is isomorphic to $T_I G$, where I is the identity element. The differential of L_g is an isomorphism between T_I and T_g. In particular, given $v \in T_I G$, we have that $g \exp(v) \in T_g G$. Therefore, it is sufficient to study only one tangent space, namely $T_I G$ that is the tangent space at the identity element. This space is usually referred to as the Lie algebra, since it can be endowed with an additional operation (the Lie bracket) which makes it indeed an algebra. When we consider Lie groups of matrices, the Lie algebra is again a set of matrices and the map $\exp(v)$ is simply the matrix exponential, i.e.

$$\exp(v) = \exp_m(v) = \sum_{n=0}^{\infty} \frac{v^n}{n!}$$

which is easily approximated by a variety of numerical algorithms.

8.B Proofs

8.B.1 Proof of Proposition 8.1

Proof. Let $H = 1$, then the Lévy-Khintchine representation entails

$$\begin{aligned}\mathbb{E}\left[\exp\left(-\int_\Theta f(x)\widetilde{\mu}_j(\mathrm{d}x)\right)\right] = \mathbb{E}\left[\exp\left(-\int_\Theta f(x)\lambda_{j1}\mu_h^*(\mathrm{d}x)\right)\right] = \\ \exp\left(-\int_{\mathbb{R}^+\times\Theta} (1-\exp(-s\lambda_{j1}f(x)))\,\rho_h^*(s)\mathrm{d}s\,\alpha_h^*(\mathrm{d}x)\right) = \\ \exp\left(-\int_{\mathbb{R}^+\times\Theta} \left(1-\exp(-s'f(x))\right)\rho_h^*(s'/\lambda_{j1})\lambda_{j1}^{-1}\mathrm{d}s'\,\alpha_h^*(\mathrm{d}x)\right)\end{aligned}$$

where the last equality follows from the change of variables $s' = \lambda_{j1}s$. This proves the claim when $H = 1$.

In the more general case $H > 1$, we have that $\widetilde{\mu}_j$ is the superposition of the random measures $\lambda_{j1}\mu_1^*, \ldots, \lambda_{jH}\mu_H^*$, which are independent since the μ_h^*'s are. Hence, the Lévy intensity of $\widetilde{\mu}_j$ is the sum of the intensities of the $\lambda_{jh}\mu_h^*$'s. □

8.B.2 The latent factor model is not completely random

From representation (8.2) it is easy to see that $\widetilde{\mu}_1, \ldots, \widetilde{\mu}_g$ is not a vector of completely random measures. Indeed, for any two disjoint measurable sets A, B the random variables defined as

$$\widetilde{\mu}_j(A) = \sum_{k\geq 1} \gamma_{jk} J_k I[\theta_k^* \in A]$$

are not independent. This is due to the the scores $\gamma_{jk} = (\Lambda M)_{jk}, k = 1, \ldots,$ which are not a collection of independent random variables.

8.B.3 Proof of Theorem 8.1

We first state a technical lemma providing an alternative characterization of compound random measures.

Lemma 8.1. *Let $\pi_h : \mathbb{R}^H \to \mathbb{R}$ be the canonical projection along the h-th coordinate, i.e. $\pi_h(\boldsymbol{x}) = x_h$ for all $\boldsymbol{x} = (x_1, \ldots, x_H)$. Let N be a Poisson point process on $\Omega := (0, +\infty)^H \times (0, +\infty) \times \Theta$ such that*

$$N = \sum_{k \geq 1} \delta_{\mathbf{m}_k, z_k, x_k}$$

with intensity

$$\lambda_N(\mathrm{d}\boldsymbol{m}\mathrm{d}z\mathrm{d}x) = \prod_{h=1}^{H} f(m_h)\mathrm{d}m_h \nu^*(\mathrm{d}z)\alpha(\mathrm{d}x). \tag{8.15}$$

Then, the collection of random measures $\mu_1^*, \ldots, \mu_H^*$ defined aw

$$\mu_h^*(A) = \int_\Omega \pi_h(\boldsymbol{m}) z I[x \in A] N(\mathrm{d}\boldsymbol{m}\mathrm{d}z\mathrm{d}x) \tag{8.16}$$

for all measurable A is a compound random measures

Proof. The proof easily follows by writing explicitly (8.16) as

$$\mu_h^*(A) = \sum_{k \geq 1} m_{hk} J_k \delta_x(A),$$

observing that the points (J_k, x_k) form a Poisson point process with intensity $\nu^*(\mathrm{d}x)\alpha(\mathrm{d}x)$. Finally, from (8.15) it is clear that $m_{hk} \overset{\text{iid}}{\sim} f$. □

We are now ready to prove Theorem 8.1

Proof. Write

$$\mathbb{E}[\tilde{p}_j(A)] = \mathbb{E}\left[\frac{\widetilde{\mu}_j(A)}{\widetilde{\mu}_j(\mathbb{X})}\right] = \int_{\mathbb{R}_+} \sum_{h=1}^{H} \mathbb{E}\left[\lambda_{jh} e^{-u \sum_{k=1}^{H} \lambda_{jk} \mu_k^*(\mathbb{X})} \mu_h^*(A)\right] \mathrm{d}u$$

where the second equality follows from writing $\widetilde{\mu}_j(\cdot) = \sum_h \lambda_{jh} \mu^*(\cdot)$, the equality $t^{-1} = \int_{\mathbb{R}_+} e^{-ut}\mathrm{d}u$ and an application of Fubini's theorem. By the tower property of the expected value, we further have

$$\mathbb{E}[\tilde{p}_j(A)] = \int_{\mathbb{R}_+} \sum_{h=1}^{H} \mathbb{E}\left[\lambda_{jh} \mathbb{E}\left[e^{-u \sum_{k=1}^{H} \lambda_{jk} \mu_k^*(\mathbb{X})} \mu_h^*(A) \mid \Lambda\right]\right].$$

Let us consider the inner expected value. Using (8.16) we can write

$$\mathbb{E}\left[e^{-u \sum_{k=1}^{H} \lambda_{jk} \mu_k^*(\mathbb{X})} \mu_h^*(A) \mid \Lambda\right] = \mathbb{E}\left[\int_\Omega g(\boldsymbol{m}, z, x, N) N(\mathrm{d}\boldsymbol{m}\mathrm{d}z\mathrm{d}x)\right]$$

where

$$g(\boldsymbol{m}, z, x, N) = e^{-u \sum_{k=1}^{H} \lambda_{jk} \mu_k^*(\mathbb{X})} \pi_h(\boldsymbol{m}) z I[x \in A].$$

Observe further that, although not explicitly written, $\mu_k^*(\mathbb{X})$ is of course a function of N. By the Campbell-Little-Mecke formula,

$$E\left[e^{-u\sum_{k=1}^H \lambda_{jk}\mu_k^*(\mathbb{X})}\mu_h^*(A)\,|\,\Lambda\right] = \int_\Omega g(\boldsymbol{m}, z, x, N+\delta_{(\mathbf{m},z,x)})\lambda_N(\mathrm{d}\boldsymbol{m}\mathrm{d}z\mathrm{d}x)$$

where λ_N is as in (8.16). Focusing on the integrand, we have

$$g(\boldsymbol{m}, z, x, N+\delta_{(\mathbf{m},z,x)}) = e^{-u\sum_{k=1}^H \lambda_{jk}(\mu_k^*+\pi_k(\mathbf{m})z\delta_x)(\mathbb{X})}\pi_h(\boldsymbol{m})zI[x\in A].$$

With an abuse of notation, let us denote with f the probability density of the $m'_{hk}s$, so that

$$\begin{aligned}
&E\left[e^{-u\sum_{k=1}^H \lambda_{jk}\mu_k^*(\mathbb{X})}\mu_h^*(A)\,|\,\Lambda\right]\\
&\quad= \int_\Omega \mathbb{E}\left[e^{-u\sum_{k=1}^H \lambda_{jk}\mu_k^*(\mathbb{X})}\,|\,\Lambda\right]\prod_{k=1}^H e^{-u\lambda_{jk}m_k z}m_h zI[x\in A]\prod_{k=1}^H f(m_k)\mathrm{d}m_k\nu^*(\mathrm{d}z)\alpha(\mathrm{d}x)\\
&\quad= \alpha(A)\mathbb{E}\left[e^{-u\sum_{k=1}^H \lambda_{jk}\mu_k^*(\mathbb{X})}\,|\,\Lambda\right]\int_{\mathbb{R}_+} z\prod_{k\neq h}\int_{\mathbb{R}_+} e^{-u\lambda_{jk}m_k z}f(m_k)\mathrm{d}m_k\\
&\qquad\qquad\times\int_{\mathbb{R}_+} e^{-u\lambda_{jh}m_h z}m_h f(m_h)\mathrm{d}m_h\nu^*(\mathrm{d}z)\\
&\quad= \alpha(A)\mathbb{E}\left[e^{-u\sum_{k=1}^H \lambda_{jk}\mu_k^*(\mathbb{X})}\,|\,\Lambda\right]\int_{\mathbb{R}_+} z\prod_{k\neq h}\mathcal{L}(u\lambda_{jk}z)\kappa(u\lambda_{jh}z,1)\nu^*(\mathrm{d}z)\\
&\quad= \alpha(A)\psi_\rho(u\lambda_{j1},\dots,u\lambda_{jH})\int_{\mathbb{R}_+} z\prod_{k\neq h}\mathcal{L}_f(u\lambda_{jk}z)\kappa_f(u\lambda_{jh}z,1)\nu^*(\mathrm{d}z)
\end{aligned}$$

where ψ_ρ is the Laplace transform of $(\mu_1^*,\dots,\mu_H^*)$ evaluated at the constant functions $u\lambda_{j1},\dots,u\lambda_{jH}$, $\mathcal{L}_f$ denotes the Laplace transform of the density f and $\kappa_f(x,n) := \int e^{-x}m^n f(m)\mathrm{d}m$.

Hence,

$$\mathbb{E}[\tilde{p}_j(A)] = \alpha(A)\sum_{h=1}^H\int\mathbb{E}\left[\lambda_{jh}\psi_\rho(u\lambda_{j1},\dots,u\lambda_{jH})\int_{\mathbb{R}_+} z\prod_{k\neq h}\mathcal{L}_f(u\lambda_{jk}z)\kappa_f(u\lambda_{jh}z,1)\nu^*(\mathrm{d}z)\right]\mathrm{d}u$$

□

8.B.4 Proof of Proposition 8.2

$$\begin{aligned}
\mathrm{Cov}\left[\widetilde{\mu}_j(A),\widetilde{\mu}_\ell(B)\right] &= \mathrm{Cov}\left[\sum_{h=1}^H\lambda_{j,h}\mu_h^*(A),\sum_{k=1}^H\lambda_{\ell,k}\mu_k^*(B)\right]\\
&= \mathbb{E}\left[\sum_{h,k}\left(\lambda_{jh}\mu_h^*(A)-\bar{\lambda}_{jh}m_h^*(A)\right)\left(\lambda_{\ell k}\mu_k^*(B)-\bar{\lambda}_{\ell k}m_k^*(B)\right)\right]\\
&= \sum_{h,k}\mathbb{E}\left[\lambda_{jh}\lambda_{\ell_k}\mu_h^*(A)\mu_k^*(B)\right]-\mathbb{E}[\lambda_{jh}\mu_h^*(A)]\bar{\lambda}_{\ell k}m_k^*(B)+\\
&\qquad\qquad-\bar{\lambda}_{jh}m_k^*(A)\mathbb{E}[\lambda_{\ell k}\mu_k^*(B)]+\bar{\lambda}_{jh}\bar{\lambda}_{\ell k}m_k^*(A)m_k^*(B)\\
&= \sum_{h,k}\mathbb{E}[\lambda_{jh}\lambda_{\ell k}]\mathbb{E}[\mu_h^*(A)\mu_k^*(B)]-\bar{\lambda}_{jh}\bar{\lambda}_{\ell k}m_k^*(A)m_k^*(B)
\end{aligned}$$

In the most general case, we thus have that

$$\begin{aligned}
\mathrm{Cov}\,[\widetilde{\mu}_j(A), \widetilde{\mu}_\ell(B)] &= \sum_h \mathbb{E}[\lambda_{jh}\lambda_{\ell h}]\mathbb{E}[\mu^*_h(A)\mu^*_h(B)] - \bar{\lambda}_{jh}\bar{\lambda}_{\ell h} m^*_k(A)m^*_h(B) + \\
&\quad \sum_{h\neq k} \mathbb{E}[\lambda_{jh}\lambda_{\ell k}]\mathbb{E}[\mu^*_h(A)\mu^*_k(B)] - \bar{\lambda}_{jh}\bar{\lambda}_{\ell k} m^*_k(A)m^*_k(B) \\
&= \sum_h \mathbb{E}[\lambda_{jh}\lambda_{\ell h}]\mathrm{Cov}(\mu^*_h(A), \mu^*_h(B)) + \mathrm{Cov}(\lambda_{jh}, \lambda_{\ell h}) m^*_h(A)m^*_h(B) + \\
&\quad \sum_{h\neq k} \mathbb{E}[\lambda_{jh}\lambda_{\ell k}]\mathrm{Cov}(\mu^*_h(A), \mu^*_k(B)) + \mathrm{Cov}(\lambda_{jh}, \lambda_{\ell k}) m^*_h(A)m^*_k(B)
\end{aligned}$$

8.B.5 Covariances and Correlations

The case of Gamma$(\Psi, 1)$ scores. Specializing Proposition 8.2 we have

$$\begin{aligned}
\mathrm{Cov}\,[\widetilde{\mu}_j(A), \widetilde{\mu}_\ell(A)] &= \mathbb{E}[\mu^*_1(A)^2]H\psi^2 + (c_A + m_A^2)H(H-1)\psi^2 - m_A^2 H^2\psi^2 \\
&= (\mathrm{Var}[\mu^*_1(A)]H + c_A H(H-1))\psi^2
\end{aligned}$$

Moreover,

$$\begin{aligned}
\mathrm{Var}[\widetilde{\mu}_j(A)] &= \mathbb{E}[\mu^*_1(A)^2]H\psi(\psi+1) + (c_A + m_A^2)H(H-1)\psi^2 - m_A^2 H^2\psi^2 \\
&= (\mathrm{Var}[\mu^*_1(A)]H + c_A H(H-1))\psi^2 + \mathbb{E}[\mu^*_1(A)^2]H\psi
\end{aligned}$$

Simple algebra leads to Equation (8.6)

The multiplicative gamma process case. Using standard properties of inverse-gamma distributed random variables, we get

$$\begin{aligned}
&\mathrm{Cov}\,[\widetilde{\mu}_j(A), \widetilde{\mu}_\ell(A)] = \\
&\quad \mathbb{E}[\mu^*_1(A)^2]\left(\sum_{h=1}^{H}(a_2-1)^{-h+1}(a_2-2)^{-h+1}\right)(a_1-1)^{-1}(a_1-2)^{-1}\left(\frac{\nu}{\nu-2}\right)^2 \\
&\quad + (c_A + m_A^2)\left(2\sum_{h<k}(a_2-1)^{-k+1}(a_2-2)^{-h+1}\right)(a_1-1)^{-1}(a_1-2)^{-1}\left(\frac{\nu}{\nu-2}\right)^2 \\
&\quad - m_A^2\left(\sum_{h,k}(a_2-1)^{-h-k+1}\right)(a_1-1)^{-2}\left(\frac{\nu}{\nu-2}\right)^2
\end{aligned}$$

and

$$\begin{aligned}
&\mathrm{Var}[\widetilde{\mu}_j(A)] = \\
&\quad \mathbb{E}[\mu^*_1(A)^2]\left(\sum_{h=1}^{H}(a_2-1)^{-h+1}(a_2-2)^{-h+1}\right)(a_1-1)^{-1}(a_1-2)^{-1}\frac{\nu^2}{(\nu-2)(\nu-4)} \\
&\quad + (c_A + m_A^2)\left(2\sum_{h<k}(a_2-1)^{-k+1}(a_2-2)^{-h+1}\right)(a_1-1)^{-1}(a_1-2)^{-1}\left(\frac{\nu}{\nu-2}\right)^2 \\
&\quad - m_A^2\left(\sum_{h,k}(a_2-1)^{-h-k+1}\right)(a_1-1)^{-2}\left(\frac{\nu}{\nu-2}\right)^2
\end{aligned}$$

Note that the only term differing in the expressions of $\mathrm{Cov}\,[\widetilde{\mu}_j(A), \widetilde{\mu}_\ell(A)]$ and $\mathrm{Var}[\widetilde{\mu}_j(A)]$ is the last factor in the first row.

8.B.6 Proof of Propsition 8.3

The first point follows directly from the definition of the gamma process. Regarding the second one, we recall a general expression given in Griffin and Leisen (2017).

Theorem 8.3. *[*Mixed moments of CoRMs*, (Theorem 6, Griffin and Leisen, 2017)] Let $q_h \geq 0$, $i = h, \dots, H$ such that $\sum_h q_h = k$. Then*

$$\mathbb{E}\left[\prod_{h=1}^{H} (\mu_h^*(A)^{q_h})\right] = \prod_h q_h! \left(\sum_{j=1}^{k} \alpha(A)^{\ell}\right)$$
$$\times \sum_{j=1}^{k} \sum_{\boldsymbol{\eta}, \boldsymbol{s}_1, \dots, \boldsymbol{s}_j \in p_j(k)} \prod_{i=1}^{j} \frac{1}{\eta_i!} \left[\prod_{h=1}^{H} \frac{(\phi)_{s_{hi}}}{s_{hi}!} \int z^{s_{1i} + \dots + s_{Hi}} \nu^*(\mathrm{d}z)\right]^{\eta_j} \tag{8.17}$$

where $p_j(k)$ is the set of vectors $(\boldsymbol{\eta}, \boldsymbol{s}_1, \dots, \boldsymbol{s}_j)$, $\eta = (\eta_1, \dots, \eta_j)$, $\boldsymbol{s}_i = (s_{i1}, \dots, s_{iH})$, such that η_i is positive, $\sum \eta_i = k$, $\mathbf{0} \prec \boldsymbol{s}_1 \prec \dots \prec \boldsymbol{s}_j$ and $\sum_{i=1}^{j} \eta_i(s_{i1} + \dots + c_{Hi}) = k$.

It suffices to consider the case $\boldsymbol{q} = (1, 1, 0, \dots, 0)$. Then, the problem consists in understanding how the sets $p_j(2)$ are made for $j = 1, 2$. The only possible vector $\boldsymbol{\eta}$ in $p_1(2)$ is $\boldsymbol{\eta} = (2)$. Therefore the only possible $\boldsymbol{s}_1$ is $\boldsymbol{s}_1 = (1, 0, \dots, 0)$. Hence the sum over $\boldsymbol{\eta}, \boldsymbol{s}_1, \dots, \boldsymbol{s}_j \in p_j(k)$ when $j = 1$ equals to

$$\frac{1}{2}\left[\phi \int z\nu^*(\mathrm{d}z)\right]^2$$

When $j = 2$, we have that the possible $\boldsymbol{\eta}$'s are $(0, 2)$, $(1, 1)$, $(2, 0)$. Note that the first and last candidate cannot satisfy $\sum_{i=1}^{j} \eta_i(s_{i1} + \dots + c_{Hi}) = k$ for any choice of $\boldsymbol{s}$. Therefore, we can consider $\boldsymbol{\eta} = (1, 1)$, leading to $\boldsymbol{s}_1 = (0, 1, 0, \dots, 0)$ and $\boldsymbol{s}_2 = (1, 0, \dots, 0)$. Hence the sum over $\boldsymbol{\eta}, \boldsymbol{s}_1, \dots, \boldsymbol{s}_j \in p_j(k)$ when $j = 2$ equals to

$$\phi^2 \left[\int z\nu^*(\mathrm{d}z)\right] \left[\int z\nu^*(\mathrm{d}z)\right]$$

Finally, observe that when the CoRM has gamma marginals, $\int z\nu^*(\mathrm{d}z) = B(1, \phi)$, where B is the Beta function. This concludes the proof.

8.B.7 Proof of Proposition 8.4

Let $\{E_n\}_n$ be the generators for $\mathfrak{sl}(H)$. Then

$$\Pi_{\mathfrak{sl}(H)} = \sum_n \mathrm{tr}(XE_n)E_n$$

It is easy to see that such a set of generators is given by:

$$\left\{\bigcup_{\ell \neq m} A : A_{ij} = \delta_{\ell,m}(i, j)\right\} \cup \left\{\bigcup_{\ell=1}^{H-1} A : A_{i,i} = 1, A_{i+1,i+1} = -1\right\}$$

which consists of $H(H-1)$ (first term) plus $H-1$ (second term) elements. We call the two sets above A_1^* and A_2^* respectively.

For numerical purposes, we don't need to compute the inner product and sum with all the $H^2 - 1$ elements in the basis. In fact note that when $E_n \in A_1^*$, say E_n is nonzero only in element i, j, $\mathrm{tr}(XE_n)E_n$ is a matrix whose only nonzero entry is the j, i-th with value $X_{i,j}$. Therefore

$$\sum_{E_n \in A_1^*} \mathrm{tr}(XE_n)E_n = (X - \mathrm{diag}(X))^T,$$

where $\mathrm{diag}(X)$ is the diagonal matrix with entries equal to the diagonal of X. Similarly, when $E_n \in A_2^*$, $\mathrm{tr}(XE_n)E_n$ is a diagonal matrix whose nonzero entries are the (i,i)-th and $(i+1, i+1)$-th and are equal to $\pm X_{i,i} - X_{i+1,i+1}$ respectively.

8.C Slice Sampling Algorithm

Let $T_j = \sum_{\ell \geq 1} (\Lambda M)_{j\ell} J_\ell$ and introduce auxiliary cluster allocation variables $c_{j,i}$ (one for each observation $y_{j,i}$) as well as auxiliary latent variables U_j such that $U_j \,|\, T_j \sim \mathrm{Gamma}(n_j, T_j)$. Standard computations lead to the extended likelihood

$$p(\{y_{j,i}\}, \{c_{j,i}\}, \{u_j\} \mid \cdots) = \left[\prod_{j=1}^{g} \frac{1}{\Gamma(n_j)} u_j^{n_j - 1} \right] \times \\ \prod_{j=1}^{g} \prod_{i=1}^{n_j} f(y_{j,i} \,|\, \theta_{c_{j,i}}) (\Lambda M)_{j, c_{j,i}} J_{c_{j,i}} \times \exp\left(- \sum_{j=1}^{g} u_j \sum_{\ell=1}^{\infty} (\Lambda M)_{j,\ell} J_\ell \right)$$

We further introduces auxiliary slice variables $s_{j,i}$ so that

$$p(\{y_{j,i}\}, \{c_{j,i}\}, \{u_j\} \mid \cdots) = \left[\prod_{j=1}^{g} \frac{1}{\Gamma(n_j)} u_j^{n_j - 1} \right] \times \\ \prod_{j=1}^{g} \prod_{i=1}^{n_j} f(y_{j,i} \,|\, \theta_{c_{j,i}}) (\Lambda M)_{j, c_{j,i}} I(s_{j,i} < J_{c_{j,i}}) \times \exp\left(- \sum_{j=1}^{g} u_j \sum_{\ell=1}^{\infty} (\Lambda M)_{j,\ell} J_\ell \right)$$

where $I(\cdot)$ denotes the indicator function. Then, we can devide between *active* and *non-active* components: let $L = \min s_{j,i}$, $J^a = \{J_\ell \text{ s.t. } J_\ell > L\}$ and $J^{na} = J \setminus J^a$, we further denote with k the cardinality of J^a, observe that k is finite almost suerly. Analogously define M^a the $H \times k$ matrix with columns $\{m_\ell \text{ s.t. } J_\ell > L\}$ and M^{na} in a similar fashion. The likelihood can be rewritten as

$$p(\{y_{j,i}\}, \{c_{j,i}\}, \{u_j\} \mid \cdots) = \left[\prod_{j=1}^{g} \frac{1}{\Gamma(n_j)} u_j^{n_j - 1} \right] \times \\ \prod_{j=1}^{g} \prod_{i=1}^{n_j} f(y_{j,i} \,|\, \theta_{c_{j,i}}) (\Lambda M)_{j, c_{j,i}} I(s_{j,i} < J_{c_{j,i}}) \times \exp\left(- \sum_{j=1}^{g} u_j \sum_{\ell=1}^{k} (\Lambda M^a)_{j,\ell} J^a_\ell \right) \\ \exp\left(- \sum_{j=1}^{g} u_j \sum_{\ell=1}^{\infty} (\Lambda M^{na})_{j,\ell} J^{na}_\ell \right)$$

To compute posterior inference, we need to be able to marginalize over M^{na} and J^{na}, and compute

$$\mathbb{E}\left[\exp\left(- \sum_{j=1}^{g} u_j \sum_{\ell=1}^{\infty} (\Lambda M^{na})_{j,\ell} J^{na}_\ell \right) \Big| \Lambda \right] \tag{8.18}$$

We manipulate the sum in the exponential to get

$$\begin{aligned}
\exp\left(-\sum_{j=1}^{g} u_j \sum_{\ell=1}^{\infty} (\Lambda M^{na})_{j,\ell} J_\ell^{na}\right) &= \exp\left(-\sum_{j=1}^{g} u_j \sum_{\ell=1}^{\infty}\sum_{h=1}^{H} \lambda_{j,h} m_{h,\ell}^{na} J_\ell^{na}\right) \\
&= \exp\left(-\sum_{h=1}^{H}\sum_{j=1}^{g} u_j \lambda_{j,h} \sum_{\ell=1}^{\infty} m_{h,\ell}^{na} J_\ell^{(na)}\right) \\
&= \exp\left(-\sum_{h=1}^{H}\left[\left(\sum_{j=1}^{g} u_j\lambda_{j,h}\right)\left(\sum_{\ell=1}^{\infty} m_{h,\ell}^{na} J_\ell^{(na)}\right)\right]\right)
\end{aligned}$$

So that (8.18) can be computed by virtue of Theorem 1 in Griffin and Leisen (2017), replacing v_j (in their notation) with $\sum_{j=1}^{g} u_j \lambda_{j,h}$.

Then, the MCMC algorithm follows the same lines of the slice sampling algorithm in Griffin and Leisen (2017).

8.D Aligning densities in higher-dimension

Computing the L_2 distance between functions is easy when the dimension of the data space is small, which is always the case in our simulations. In higher dimensional settings, we suggest instead the following dissimilarity function

$$d(\hat{\mu}, \mu')^2 = \inf_{T \in \Gamma(\hat{\mu}, \mu')} \sum_{h,k=1}^{K} W_2^2(f(\cdot \mid \hat{\theta}_h^*), f(\cdot \mid \theta'_k)) T_{hk}$$

where $\{\hat{\theta}_h^*\}_h$ and $\{\theta'_k\}_k$ are the atoms in $\hat{\mu}$ and μ' respectively, $\Gamma(\hat{\mu}, \mu')$ denotes all the $K \times K$ matrices whose row-sums are equal to the normalized weights in $\hat{\mu}$ and the column-sums are equal to the normalized weights in μ'.

That is, the distance corresponds to the Wasserstein distance between two atomic probability measures. The associated ground cost is $W_2^2(f(\cdot \mid \hat{\theta}_h^*), f(\cdot \mid \theta'_k))$ that is the squared Wasserstein distance between the probability measure with density $f(\cdot \mid \hat{\theta}_h^*)$ and the one with density $f(\cdot \mid \theta'_k)$. This choice of ground cost ensures that the specific choice of the kernel density f is taken into account.

In particular, $W_2^2(f(\cdot \mid \hat{\theta}_h^*), f(\cdot \mid \theta'_k))$ can be easily computed for location-scatter families of probability densities. Let $\mathcal{L}$ denote a generic law of a random variable, and X_0 a d-dimensional random vector with law P_0 such that $\mathbb{E}[\|X\|^2] < +\infty$ and P_0 is absolutely continuous with respect to the d-dimensional Lebesgue measure. Then a location-scatter family is the set of random variables

$$\{\mathcal{L}(\Sigma^{1/2} X_0 + \mu), \text{ such that } \Sigma \text{ is symmetric and positive definite}, \mu \in \mathbb{R}^d\}$$

This definition obviously encompasses the popular Gaussian density but also the Student-t, Laplace, and discrete and continuous uniform distributions among others.

Let $f(\cdot \mid \mu_i, \Sigma_i)$, $i = 1, 2$ denote the densities of two random variables in the location-scatter family under consideration. Theorem 2.1 and Corollary 3.12 in Álvarez-Esteban et al. (2018) entail that

$$W_2^2\left(f(\cdot \mid \mu_1, \Sigma_1), f(\cdot \mid \mu_2, \Sigma_2)\right) = \|\mu_1 - \mu_2\|^2 + \text{trace}\left(\Sigma_1 + \Sigma_2 - 2(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}\right).$$

Hence, the proposed distance can be computed exactly. The main computational bottleneck is the computation of the matrix square root. Its exact computation requires

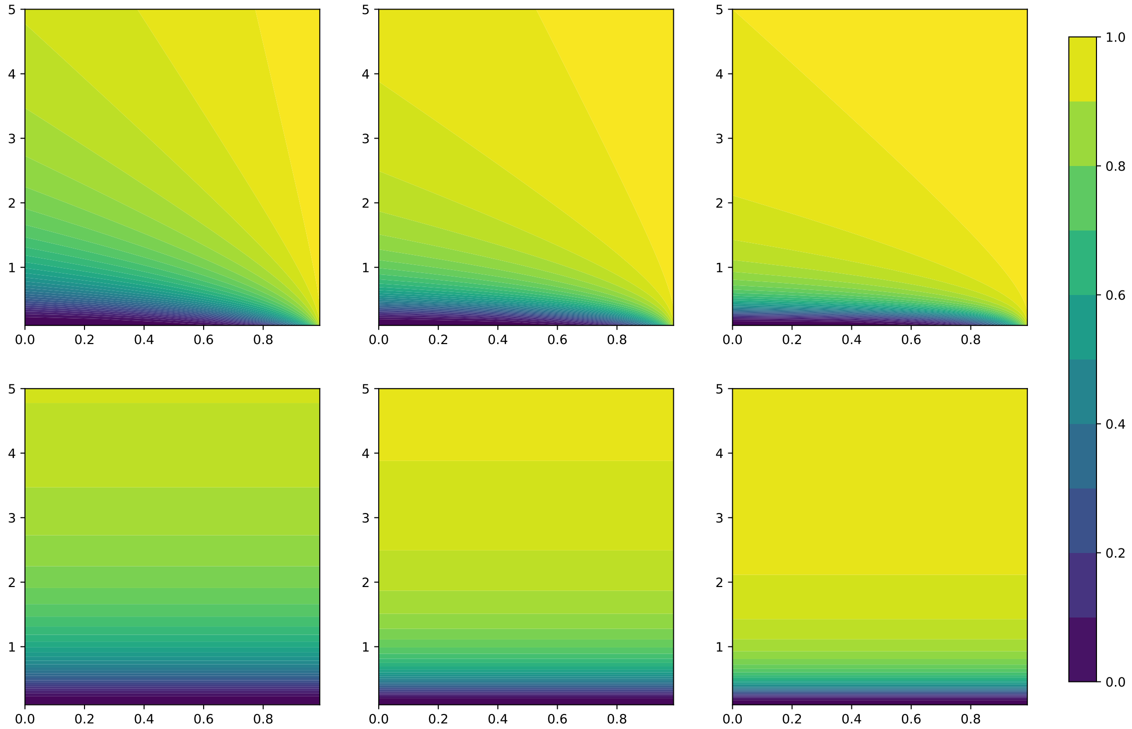


Figure 8.E.1: Correlation between neighboring $\widetilde{\mu}_i(A)$ and $\widetilde{\mu}_j(A)$ (top row) and between disconnected $\widetilde{\mu}_i(A)$ and $\widetilde{\mu}_\ell(A)$ for a set A such that $\alpha(A) = 0.5$ under prior (8.4). From left to right $H = 4, 8, 16$. The values of ρ vary across the x-axis in each plot, the values of τ across the y-axis.

computing the eigendecomposition of the matrix, whose computational cost scales cubically with the dimension. Otherwise, several approximate iterative algorithms have been proposed.

8.E Additional Simulations and Plots

Figure 8.E.1 shows the correlation between $\widetilde{\mu}_j(A)$ and $\widetilde{\mu}_\ell(A)$ under prior (8.4) above. We consider a simple setting with three areas i, j, ℓ such that areas i and j are neighboring while area ℓ is not connected to either i and j.

Figure 8.E.2 shows the variance of the ratio $r^k_{j\ell}$ defined in Equation (8.7) under different priors for Λ. As expected, the variance quickly drops to zero when the λ_{jh}'s are i.i.d. as H increases. The same happens when we assume that Λ follows a shrinkage prior, but the decay is slower.

Figure 8.E.3 shows the effect of the a priori variance of the λ_{jh}'s on the variance of $r^k_{j\ell}$.

Figure 8.E.4 shows the dendrogram of the hierarchical clustering on the rows of Λ' on the Invalsi dataset.

Figure 8.E.5 shows some exploratory data analysis for the US income dataset analyzed in Section 8.6.2.

Figure 8.E.6 shows the estimates of the latent factor measures for the US income dataset after the post-processing.

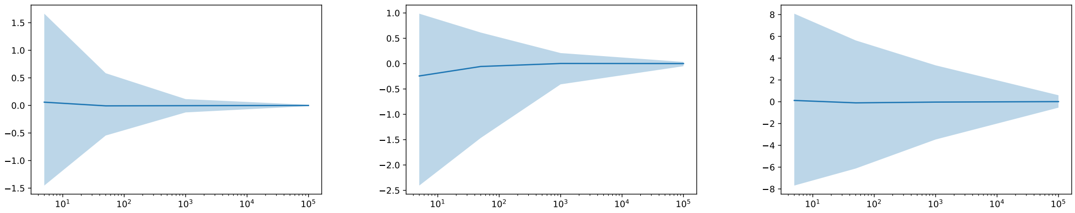


Figure 8.E.2: Monte Carlo estimate of $\log r_{j\ell}$ as a function of H under different priors: from left to right, $\lambda_j h \overset{\text{iid}}{\sim} \text{Ga}(1,1)$, $\boldsymbol{\lambda}_j = (\lambda_{j1},\dots,\lambda_{jH}) \overset{\text{iid}}{\sim} \text{MGP}(2,1)$, $\boldsymbol{\lambda}_j \overset{\text{iid}}{\sim}$ CUSP. The solid line represents the Monte Carlo average over $1,000$ simulations. The shaded area are 95% confidence intervals.

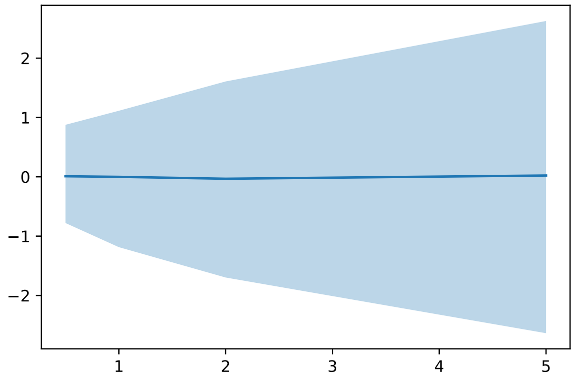


Figure 8.E.3: Monte Carlo estimate of $\log r_{j\ell}$ when λ_{jh} are i.i.d gamma variables with mean equal to 1 and increasing variance (x-axis).

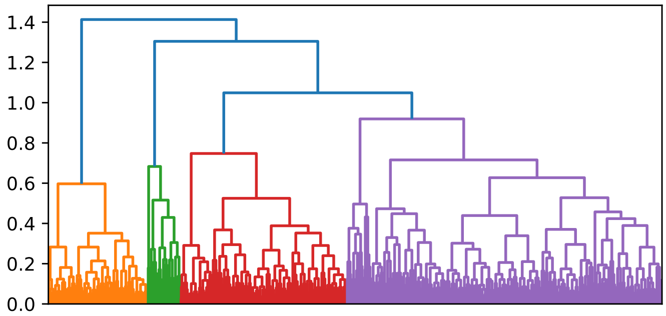


Figure 8.E.4: Dendrogram for the hierarchical clustering with complete linkage on the rows of Λ' on the Invalsi dataset.

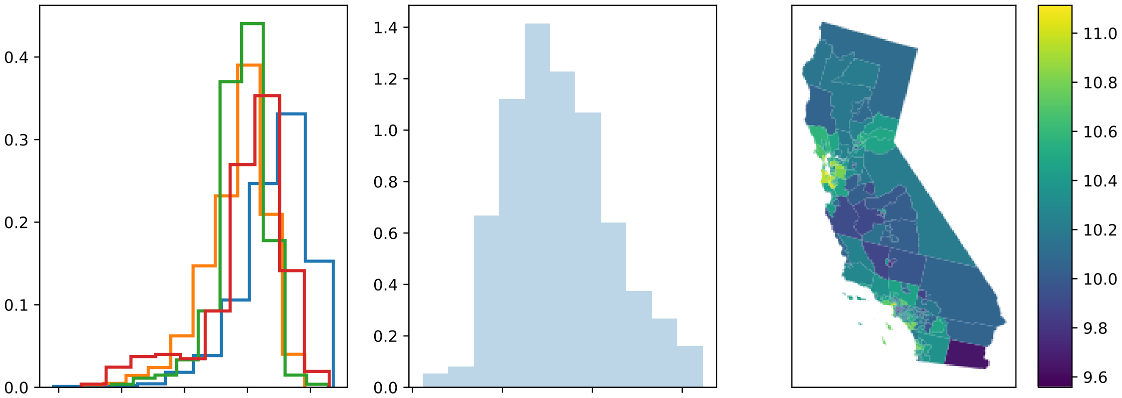


Figure 8.E.5: From left to right: histogram of the (log) incomes in five randomly sampled PUMAs, histogram of the average (log) income across all the PUMAs, average (log) income displayed in a map

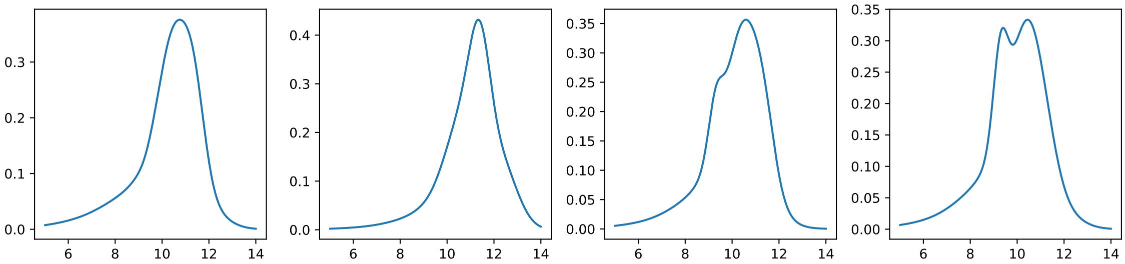


Figure 8.E.6: Estimates of $\int_\Theta f(y \mid \theta)\mu'_h(\mathrm{d}\theta)/\mu'_h(\Theta)$ after post-processing in the US Income example.

9. Bayesian Nonparametric Vector Autoregressive Models via a Logit Stick-breaking Prior an Application to Child Obesity

In this chapter, based on Beraha et al. (2022), we discuss an application of nonparametric Bayesian modelling for time series of child growth curves. It is well known that overweight and obesity in adults are known to be associated with risks of metabolic and cardiovascular diseases. Because obesity is an epidemic, increasingly affecting children, it is important to understand if this condition persists from early life to childhood and if different patterns of obesity growth can be detected. Our motivation starts from a study of obesity over time in children from South Eastern Asia. Our main focus is on clustering obesity patterns after adjusting for the effect of baseline information. Specifically, we consider a joint model for height and weight patterns taken every 6 months from birth. We propose a novel model that facilitates clustering by combining a vector autoregressive sampling model with a dependent logit stick-breaking prior. Simulation studies show the superiority of the model to capture patterns, compared to other alternatives. We apply the model to the motivating dataset, and discuss the main features of the detected clusters. We also compare alternative models with ours in terms of predictive performances.

9.1 Introduction

Overweight and obesity are defined as abnormal or excessive fat accumulation that may impair health (WHO, 2022). It is well-known that overweight and obesity in adults are associated with risks of metabolic and cardiovascular diseases; see, for instance, Després et al. (2008), Fox et al. (2007) and Pi-Sunyer (2009). Furthermore, individuals who are obese and contracted COVID-19 have an increased likelihood to experience a more severe course of illness (Gao et al., 2020).

Obesity is an epidemic, increasingly affecting children. In 2018, 18% of children in the United States were obese and approximately 6% were severely obese (Hales et al., 2018). Prevalence of obesity in children has increased from 4% in 1975 to over 18% in 2016 among children and adolescents aged 5-19 years [WHO, Accessed: 01-06-2021]; see also Cremaschi et al. (2021). Overweight or obesity in childhood is critical as it often persists into adulthood due to both physiological and behavioural factors, e.g. (i) adults diet based on energy-dense foods that are high in fat and sugars and (ii) adult physical inactivity due to the sedentary nature of many forms of work, changing modes of transportation, and increasing urbanization. Also for childhood obesity, dietary composition and sedentary lifestyle have often been cited as main contributors. Evidence also exists for a significant role of parents' socioeconomic status and maternal prenatal health factors; see, again, Cremaschi et al. (2021).

Research on the origins of health and disease suggests that susceptibility to metabolic disease may originate early in life. Different conditions in maternal uteruses seem to influence metabolic health by altering glucose metabolism and body composition (Symonds et al., 2013; Godfrey et al., 2012). Moreover, increased adiposity have been observed in school-age children and infants (Nightingale et al., 2010; Whincup et al., 2005; Yajnik et al., 2002, 2003).

It is therefore important to understand whether obesity persists from early life to childhood and if different types of obesity growth can be detected. For instance, Zhang et al. (2019) show that rates of change in Body Mass Index (BMI) at different childhood ages are differentially associated with adult obesity. Our motivating application is the study of obesity over time in a dataset of children in South Eastern Asia (see Soh et al., 2014), taken every 6 months from birth. In particular, we consider jointly their height and weight. It is known that obesity might increase the risk of metabolic diseases, and that this risk is higher in Asian populations than in White Caucasian population (Misra and Khurana, 2011). The aim of this work is first to provide a model that is flexible enough to represent longitudinal vector responses such as the height and the weight of the children of the study. Moreover, in this application, it is also crucial to cluster children according to their obesity patterns, i.e. the longitudinal trajectories. Indeed, uncovering different types of children obesity growth patterns would identify risk subgroups, which is a desired byproduct of the analysis, and in case, largely increase the ability of developing treatments targeted for various population segments. Our approach combines modeling the obesity growth curves with the flexibility provided by the adoption of a covariate-dependent Bayesian nonparametric (BNP) mixture prior. The key idea is that model-based clustering achieved through discrete subject-specific allocation variables should be driven by prior weights depending on subject-specific covariates, to obtain more similar clusters. Specifically, we assume a vector autoregressive (VAR) model to represent obesity growth, including subject-specific VAR parameters, after adjusting for covariates (of both fixed and time-varying types) available on children as well as mothers. Thus, clustering the sample of obesity growth curves is equivalent to cluster the VAR parameters. As mentioned before, the prior that we assume for these parameters is a covariate-dependent Bayesian nonparametric prior. A preliminary analysis shows that the lag 1 autoregression assumption is a reasonable approximation, with higher order lags implying no substantial gain. This is also a simpler and more parsimonious representation than alternatives such as a mixture of multivariate Gaussian distributions. The model also includes a time-dependent mean function, or, equivalently, a time-varying covariate which does not vary with the subject.

In more detail, we assume the children-specific VAR coefficients to be independently distributed according to a truncated stick-breaking prior with weights that depend on baseline covariates. This construction induces a prior on the partition of the children in the sample. Moreover, it allows for potentially empty clusters, in which case the *number of clusters* is interpreted as the number of non-empty components in the stick-breaking representation, i.e. components to which at least one observation is assigned. A BNP approach is particularly appealing for our application, since comparison with alternative models shows that a parametric dependence structure is unable to fully capture the data complexity. Among competitors, we have also included a popular covariate-dependent prior, the linear dependent Dirichlet process (Linear-DDP); in this case, our prior will be shown to have a superior performance in terms of standard model metrics.

The dependent stick-breaking prior adopted here can be seen as a finite-dimensional version of the logistic stick-breaking process described in Ren et al. (2011). Covariate dependent random probability constructions include the probit stick-breaking process (Chung and Dunson, 2009; Rodríguez and Dunson, 2011). These Bayesian nonparametric random probability measures stem out from the seminal work by MacEachern (2000) on dependent Dirichlet processes. See a review of this and related models in Quintana et al. (2022). Covariate dependent priors for random partition were first proposed in Müller et al. (2011) and Park and Dunson (2010).

VAR models may provide a flexible and powerful representation of longitudinal data, since they allow a straightforward representation of the covariance matrix of the data themselves; see, for instance, Canova and Ciccarelli (2004) and Daniels and Pourahmadi (2002). Bayesian nonparametric methods have been successfully applied to VAR models in recent years. See Kalli and Griffin (2018) for such a model applied to single subject data, and Billio et al. (2019) and Kundu and Lukemire (2021) for multiple subject data. In Billio et al. (2019) the authors propose a Dirichlet process mixture of normal-Gamma priors on the VAR autocovariance elements, as a Bayesian-Lasso prior. Kundu and Lukemire (2021) focus on matrix-variate data, providing a class of nonparametric Bayesian VAR models, based on heterogeneous multi-subject data, that enables separate clustering at multiple scales, and result in partially overlapping clusters. The temporal trend that we have imposed in our model derives from the combination of the VAR model, the time-dependent covariates and/or the mean function of time.

We note that a valid alternative to our VAR approach consists of longitudinal data models including random and/or fixed functions in time, random effects or latent stochastic processes, or through a combination of functions and robust methods accommodating without modeling covariance structure. BNP models following this approach include, e.g. Li et al. (2010), and Quintana et al. (2016); see references therein. Daniels and Pourahmadi (2002) illustrate the general context of dynamic models representation of longitudinal data with priors for the associated covariance matrices, for which the class of VAR models constitutes a particular case. Instead Quintana et al. (2016) present a BNP model for longitudinal data that includes flexible mean functions and autoregressive covariance structures. Similarly to our proposal, their clustering is imposed on the autocorrelation structure across subjects, though cluster estimates are not part of their main inferential targets.

Our first contribution is the introduction of a Bayesian model that is able to cluster obesity growth patterns combining several characteristics such as a VAR model, a covariate-dependent BNP prior for the VAR parameters driving the clustering, and the inclusion of fixed-time and time-varying covariates in the likelihood. Our second contribution is the design of an efficient Gibbs sampling algorithm to perform posterior inference, that exploits the recent results on logit stick-breaking priors by Rigon and Durante (2021). We note that this random probability measure is represented as a finite mixture with H support points, but unlike the sparse mixture in Frühwirth-Schnatter and Malsiner-Walli (2019), (i) the weights depend on covariates and (ii) come from a stick-breaking construction, thus implying stochastic dominance of the sequence itself (for a fixed value of the covariates). As mentioned earlier, our proposal results in a flexible model, for which posterior simulation is relatively cheap to implement.

Finally, Cremaschi et al. (2021) consider a more complex model in a similar framework, i.e. they provide a joint model for multiple growth markers and metabolic associations, which allows for data-driven clustering of the children and highlights metabolic pathways involved in child obesity. Unlike our approach, they assume a joint Bayesian nonparametric random effect distribution on the parameters characterizing the longitudinal trajectories of obesity and the graph capturing the association between metabolites.

The remainder of this paper is structured as follows. Section 9.2 describes the motivating application and introduces a preliminary exploratory analysis to help building the model. In Section 9.3 we present the finite mixture of VAR models and discuss its main features. Section 9.4 summarizes the results of three simulation studies carried out to test and compare posterior inference under possible alternative model formulations. Section 9.5 presents the results from the main application; we also include predictive goodness of fit to compare with alternative models. Section 9.6 concludes the paper with a discussion. The Appendix provides further plots (Appendix 9.A) and details on the Gibbs sampler algorithm for posterior simulation (Appendix 9.B).

9.2 Child growth dataset

We focus on the analysis of obesity in children from Singapore, particularly on its evolution over time. As mentioned in the Introduction, it is relevant to understand whether obesity persists from early life to childhood. Such information is of particular relevance when designing intervention policy. Section 9.2.1 introduces the data and explains the main research questions, while Section 9.2.2 contains a short summary of the exploratory analysis carried out to highlight the main data characteristics. The exploratory analysis is crucial to drive the choice of covariates and interactions in the linear term and also to inform the modeling choices adopted later in Section 9.5.

9.2.1 Description of the dataset

We consider data from *the Growing Up in Singapore Towards healthy Outcomes* (GUSTO) study, which comprises one of the most carefully phenotyped parent-offspring cohorts with a particular focus on epigenetic observations; see Soh et al. (2014) for description of the recruited women and objectives of the cohort study. The data consist of measurements of child height (or length, depending on the child's age) in centimeters and weight in kilograms from periodic visits of 1139 children from birth to the age of seven. We consider only visits occurred every 6 months, though during the first year of life, infants were visited every 3 months. More specifically, the response vector $\boldsymbol{y}_{it} \in \mathbb{R}^2$ is given by the measurements of (*length*, *weight*) up to the 12th month of age ($t = 3$) and (*height*, *weight*) from the 18th month onwards ($t = 4, \ldots, 14$). Besides sex of the child, information is available on the mother. However, the original sample includes missing observations. More in details, 77 subjects are discarded from the analysis, because only information on the first visit (i.e. right after birth) is available. Moreover, we discard children with less than two consecutive visits, and with missing baseline covariates. This leads to a final sample size of $N = 766$. Note that we keep children with missing responses, since in our Bayesian framework it is straightforward to impute these as part of the MCMC. To this end, we simulate the missing responses from their full conditional distribution at every iteration of the algorithm. See the MCMC algorithm in Appendix 9.B.

The available baseline covariates in the dataset are:

- *age*, mother's age: it ranges from 18 to 46 years.
- *parity*: number of previous pregnancies carried to a viable gestation by the mother, ranging from 0 to 5. If parity equals to 0, the child is the first born.
- *OGTT fasting Pw26*: oral glucose tolerance test (OGTT) at 24th-26th week of pregnancy; it varies from 2.9 to 8.7 mg/dL. Mothers are tested after fasting for at least eight hours.
- *OGTT 2hour Pw26*: oral glucose tolerance test at 24th-26th week of pregnancy; it ranges from 2.9 to 15.1 mg/dL. Mothers are tested two hours after having assumed a glucose solution containing a dose of sugar.
- *ppBMI*: pre-pregnancy body mass index of the mother; values in the sample range from 14.6 to 41.3 Kg/m^2.
- *GA*: gestational age in weeks, i.e. the length of the pregnancy (from 28 to 41.4 in the dataset).

- *sex*: sex of the child.
- Mother's *ethnicity*: Chinese, Malay or Indian with proportions reflecting those characterising the Singaporean population.
- Mother's highest *education*: it is a categorical variable with three ordered levels. Level 1 corresponds to no education or primary school, level 2 corresponds either to primary school, GCE (Singapore-Cambridge general certificate of education (O-level)) or ITE NTC (institute of technical education, national technical certificate) and level 3 corresponds to university degree.

The main goal of the analysis is to understand differences in obesity growth patterns among ethnic groups via the construction of clusters of individuals exhibiting different profiles. At the same time we are also interested in assessing the effect of sex, parity and gestational age of the children on the development of obesity (Tint et al., 2016). Sex, age and parity have been reported in the medical literature as associated to neonatal adiposity. Girls are known to have greater adiposity than boys even at birth (Simon et al., 2013; Fields et al., 2009; Rodríguez et al., 2004). Increasing parity is associated with increasing neonatal adiposity in Asians as well as in Western populations (Joshi et al., 2005; Catalano et al., 1995). Gestational age and postnatal age have also been shown to be associated with increasing weight and adiposity (Simon et al., 2013; Catalano et al., 1995). Other important factors relating to the mother are the results of the glucose tolerance test and pre-pregnacy body mass index, since metabolic diseases are heritable, though they do not necessarily lead to obesity (CDS, 2018); see also, for instance, Qasim et al. (2018). Since obesity might also be related to family nutritional habits, we include in the model *education* as proxy for the family socioeconomic status.

In the next subsection we present an exploratory data analysis, which will drive the choice of interactions between the covariates described above.

9.2.2 Exploratory data analysis

The three main ethnic groups in Singapore are Chinese, Malay and Indian. Their sample frequencies in the dataset, 56%, 26% and 18%, respectively, are consistent with the overall population distribution.

In Figure 9.2.1 we plot the sample correlation of the numerical covariates. We find that the largest correlation (equal to 0.42) is between *OGTT fasting* and *OGTT 2h.*

To understand the relationship between categorical and continuous covariates, Figure 9.2.2 shows histograms of each continuous covariate, stratified by each categorical covariate level. There appears to be a linear trend between *parity* and *age*, which is to be expected, and also between *parity* and *ppBMI.* Additionally, the distribution of mother's age is concentrated on smaller values for Malay and Indian ethnicity, compared to Chinese women. No other association is detectable between categorical and continuous covariates.

In Appendix 9.A we show the unidimensional scatterplots of the responses (height and weight) at time $t = 0, 1, 2$ versus the continuous covariates, with the goal of identifying effect of these covariates, which are time-homogeneous (recorded at baseline), on the responses (time-varying). For categorical covariates we plot by boxplots of the responses stratified by level. See Figure 9.A.1-9.A.2 (in the Appendix), which display a time-increasing response patterns, though there does not seem to be a clear dependence of weight and height on the covariates.

Figure 9.2.3 shows the scatterplots of the children's height (left) and weight (right) at lag 1, i.e. we plot sample points (y_{it}, y_{it+1}) for all t and all subject i for both responses y. We identify two sub-groups in both plots, corresponding to newborns and infants (the group of datapoints on the left bottom corner) and older children. For the latter

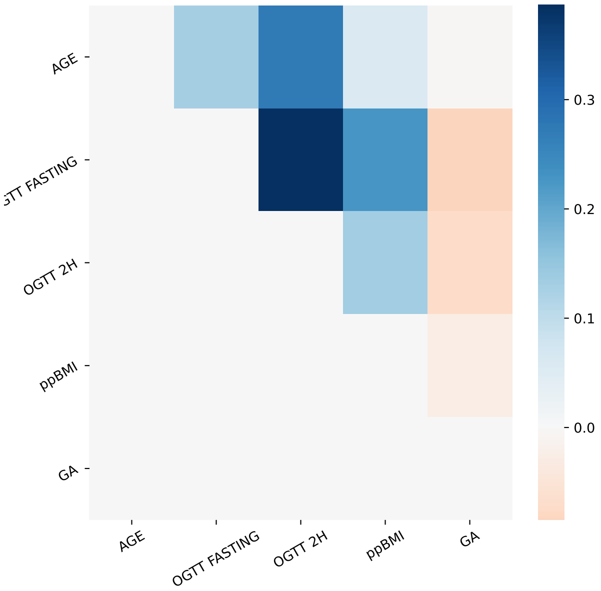


Figure 9.2.1: Sample correlation between numerical covariates in Section 9.2.1

the autoregressive assumption is very clear, while for the infant group, as expected, the linearity assumption is not strong, though it could be used as first approximation.

As such, we propose a VAR model with lag 1 for the responses. Moreover, we include in the analysis the time-homogeneous covariates $\boldsymbol{z}_i$ and a function of time, $x_{it} = \sqrt{t}$, as time-varying covariate in the model, to account for a global growth trend over time. No other time-varying covariate is available in the dataset. Other alternative mean functions of time could be considered, but this one is enough to explain the trend of weight and height in the empirical age range. We also consider interaction terms between (i) the mother's highest education and age, and (ii) ethnicity and sex of the child. Finally, denoting by $X : Y$ the interaction term between X and Y, we include the following covariates in the model: (1) an intercept, (2) *age*, (3) *parity*, (4) *OGTT fasting Pw26* (in what follows referred to as *OGTT fasting*), (5) *OGTT 2h Pw26* (in what follows referred to as *OGTT 2h*), (6) *ppBMI*, (7) *GA*, (8) *education*$_1$*:age* (9) *education*$_2$*:age*,(10) *education*$_3$*:age*, (11) *parity:age*, (12) *Indian*, an indicator variable, equal to if the mother is Indian and zero otherwise, (13) *Malay* an indicator variable, equal to 1 if the mother is Malay and zero otherwise, (14) *Male:Chinese* indicator variable equal to 1 for a male child born to a Chinese mother, (15) *Male:Indian* indicator variable equal to 1 for a male child born to an Indian mother and (16) *Male:Malay* indicator variable equal to 1 for a male child born to a Malay mother.

The baseline category for the categorical covariates corresponds to a female child born to a Chinese mother. As final pre-processing step, we standardize each numerical covariate at baseline by subtracting their sample mean and dividing by the sample standard deviation.

In summary, the Child Growth dataset contains information on $N = 766$ children, $k = 2$ responses, $p = 1$ time-dependent covariate (that is $\sqrt{t}$) and a $q = 14$-dimensional design matrix for time-homogeneous covariates (including intercepts, interactions and dummy variables to represent categorical covariates).

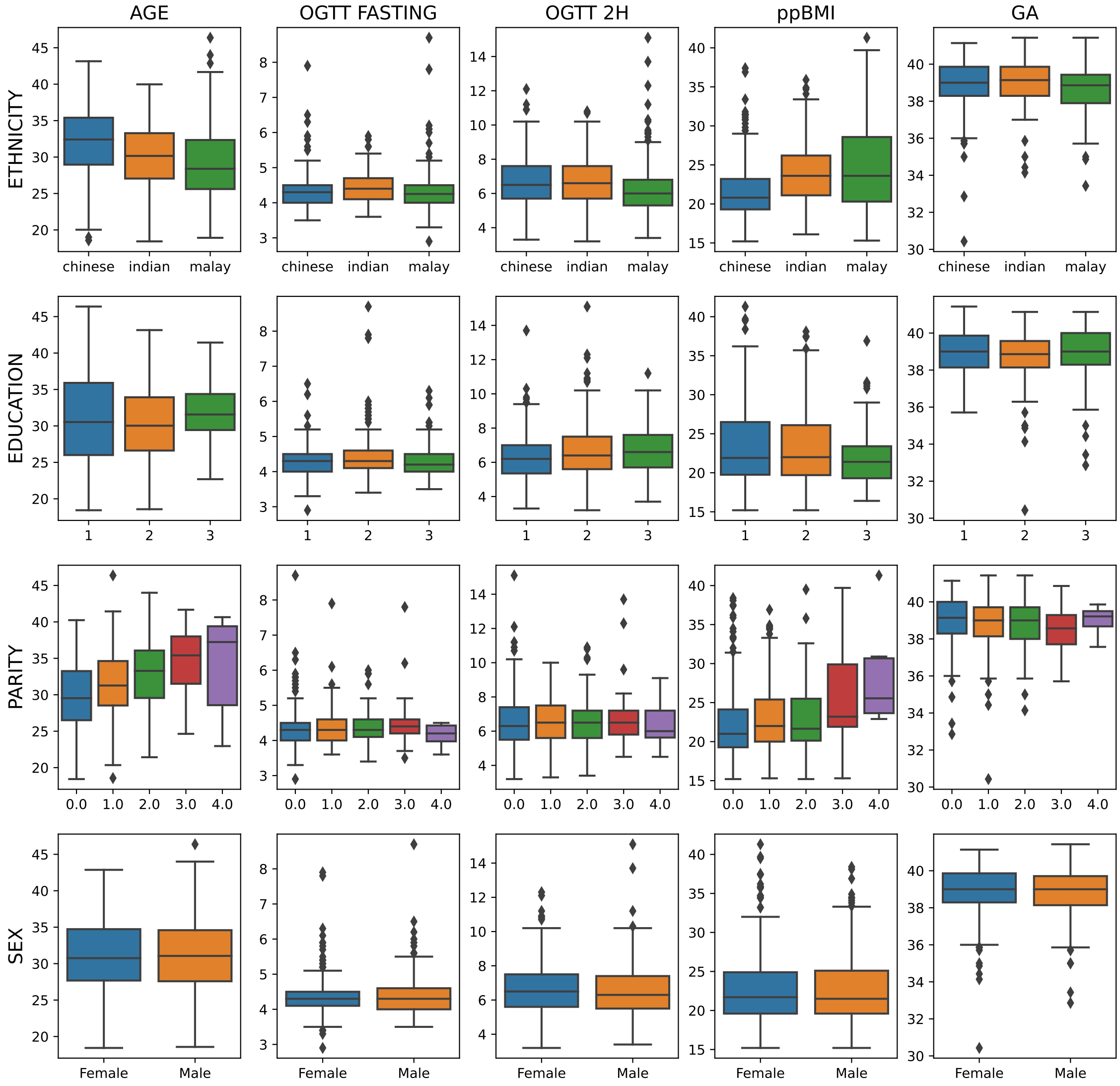


Figure 9.2.2: Boxplots of numerical variables (by column) for each level of the categorical variables (by row).

9.3 The VAR model and the logit stick-breaking prior for the VAR parameters

Our motivating application requires the development of statistical methodology able to describe the evolution of a k-dimensional response vector $\boldsymbol{Y}_{it}$ for individuals i, $i = 1, \ldots, N$ recorded at discrete time points t, $t = 1, \ldots, T_i$, accounting for time-varying covariates $\boldsymbol{x}_{it}$ and time-homogeneous covariates $\boldsymbol{z}_i$, measured at the baseline. Motivated by the exploratory analysis in Section 9.2, we assume:

$$\boldsymbol{y}_{it} = \Phi_i \boldsymbol{y}_{it-1} + B\boldsymbol{x}_{it} + \Gamma \boldsymbol{z}_i + \boldsymbol{\varepsilon}_{it},\ \boldsymbol{\varepsilon}_{it} \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0}, \Sigma),\ t = 1, \ldots, T_i,\ i = 1, \ldots, N, \qquad (9.1)$$

where $\Phi_i = [\Phi_{ijl}]$ is a $k \times k$ matrix of autoregression coefficients, $\boldsymbol{x}_{it}$ is a $p-$dimensional vector of time-varying covariates, $\boldsymbol{z}_i$ is a $q-$dimensional vector of time-homogeneous covariates, $B = [b_{jl}]$ and $\Gamma = [\gamma_{jl}]$ are $k \times p$ and $k \times q$ matrices of regression coefficients, respectively. For ease of explanation, we vectorize matrices Φ_i, B and Γ. Specifically,

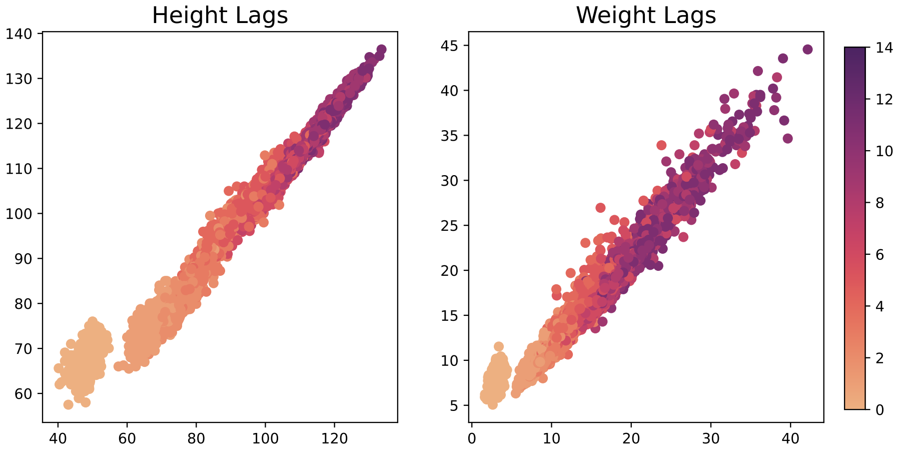


Figure 9.2.3: Scatterplots of Singapore children's height (left) and weight (right) at lag 1, i.e. of the sample points (y_{it}, y_{it+1}), for $t = 1, \dots, T_i - 1$ and $i = 1, \dots, N$ for response y; color corresponds to the age in the colorbar

denoting with $(\cdot)^T$ the transpose of a column vector, we introduce the following notation

$$\begin{aligned}\varphi_i &= (\Phi_{i11}, \dots, \Phi_{i1k}, \Phi_{i21}, \dots, \Phi_{i2k}, \dots, \Phi_{ik1}, \dots, \Phi_{ikk})^T \\ \boldsymbol{b} &= (b_{11}, \dots, b_{1p}, b_{21}, \dots, b_{2p}, \dots, b_{k1}, \dots, b_{kp})^T \\ \boldsymbol{\gamma} &= (\gamma_{11}, \dots, \gamma_{1q}, \gamma_{21}, \dots, \gamma_{2q}, \dots, \gamma_{k1}, \dots, \gamma_{kq})^T,\end{aligned}$$

so that φ_i, $\boldsymbol{b}$ and $\boldsymbol{\gamma}$ are vectors with k^2, $k \times p$ and $k \times q$ elements (vectorization of the matrices Φ_i, B, Γ, respectively). We assume $\boldsymbol{y}_{i0} = \boldsymbol{0}$, that is, conditionally to the remaining parameters, $\boldsymbol{y}_{i1}$ has a Gaussian distribution with mean $B\boldsymbol{x}_{i1} + \Gamma\boldsymbol{z}_i$. Alternatively, we could consider the responses at baseline as exogenous. Moreover, different initial distribution could be specified. We assume that a priori $(\Phi_1, \dots, \Phi_N)$, $\boldsymbol{b}$, $\boldsymbol{\gamma}$ and Σ are independent. As random effect distribution we assume a Bayesian nonparametric prior which depends on the baseline covariates. Specifically, we assume that

$$\Phi_i \mid z_i \overset{\text{ind}}{\sim} \sum_{h=1}^{H} w_h(\boldsymbol{z}_i)\delta_{\Phi_{0h}} \quad i = 1, \dots, N. \tag{9.2}$$

and we impose a stick-breaking construction on the weights w_h. As such, equation (9.2) defines a truncated stick-breaking prior with H support points $\{\Phi_{0h}\}$ and covariate-dependent weights summing to 1. Similarly to Rigon and Durante (2021), we assume that the weights are generated via a logit stick-breaking construction, that is, $w_1(\boldsymbol{z}_i) = \nu_1(\boldsymbol{z}_i)$, and $w_h(\boldsymbol{z}_i) = \nu_h(\boldsymbol{z}_i)\prod_{l=1}^{h-1}(1 - \nu_l(\boldsymbol{z}_i))$ for $h = 1, \dots, H-1$, and $\nu_H(\boldsymbol{z}_i) = 1$. The dependence on the covariates $\boldsymbol{z}_i$ is introduced by assuming a logistic model for $\nu_h(\boldsymbol{z}_i)$:

$$\begin{aligned}\text{logit}(\nu_h(\boldsymbol{z}_i)) &= \boldsymbol{z}_i^T\boldsymbol{\alpha}_h, \quad h = 1, \dots, H-1 \\ \boldsymbol{\alpha}_h &\overset{\text{iid}}{\sim} \mathcal{N}_q(\mu_\alpha, \Sigma_\alpha), \quad h = 1, \dots H-1\end{aligned} \tag{9.3}$$

An equivalent formulation of (9.2) can be obtained by introducing auxiliary variables c_i's (usually referred to as cluster allocation indicators) such that

$$c_i \mid z_i, \boldsymbol{\alpha} \sim \text{Categorical}\left(\{1, \dots, H\}; \boldsymbol{w}(\boldsymbol{z}_i)\right),$$

and letting $\Phi_i = \Phi_{0c_i}$. The introduction of the c_i's allows us to make a fundamental distinction between mixture components and clusters. In the following, we refer to any of the Φ_{0h}'s as a *component*, while we call a *cluster* of observations a (nonempty) set $\{i : c_i = h\}$; see, for instance, Argiento and De Iorio (2022). The marginal prior (9.2) - (9.3) is represented by a finite, though large number of parameters, and can be regarded as the truncation of a dependent Bayesian nonparametric prior.

We complete the prior specification with the marginal parametric prior distributions of $\boldsymbol{b}$, $\boldsymbol{\gamma}$ and Σ:

$$\boldsymbol{b} \sim \mathcal{N}_{kp}(\mathbf{0}, \Sigma_B), \qquad \boldsymbol{\gamma} \sim \mathcal{N}_{kq}(\mathbf{0}, \Sigma_\Gamma), \qquad \Sigma^{-1} \sim \mathcal{W}(\Sigma_0, \nu), \tag{9.4}$$

where $\mathcal{W}(\Sigma_0, \nu)$ denotes the Wishart distribution with expectation equal to $\nu\Sigma_0$ for $\nu > p - 1$.

To obtain more robust inference, we assume a hierarchical prior for the φ_{0h}'s:

$$\varphi_{0h} | \varphi_{00}, V_0 \overset{\text{iid}}{\sim} \mathcal{N}_{k^2}(\varphi_{00}, V_0), \qquad h = 1, \ldots, H \tag{9.5}$$

$$\varphi_{00}, V_0 | \varphi_{000}, \lambda, V_{00}, \tau_0 \sim \mathcal{NIW}(\varphi_{000}, \lambda, V_{00}, \tau_0). \tag{9.6}$$

In (9.6), $\mathcal{NIW}(\varphi_{000}, \lambda, V_{00}, \tau_0)$ denotes the normal-Inverse Wishart distribution, i.e. $V_0 \sim \mathcal{IW}(\tau_0, V_{00})$ and $\varphi_{00} \mid V_0 \sim \mathcal{N}(\varphi_{000}, \lambda^{-1} V_0)$, where $\mathcal{IW}(\tau_0, V_{00})$ denotes the inverse-Wishart distribution defined over the space of $k^2 \times k^2$ symmetric and positive definite matrices with mean $V_0/(\tau_0 - k^2 - 1)$.

Posterior inference is performed through a Gibbs sampler algorithm, as detailed in Appendix 9.B. However, it is worth noting that the full-conditional of the weights parameters $\{\boldsymbol{\alpha}_h\}$ in Equation (9.3) can be derived in closed-form with the introduction of auxiliary variables, using results in Polson et al. (2013) and Rigon and Durante (2021). The full conditional distributions of $\boldsymbol{b}$ and $\boldsymbol{\gamma}$ are derived as in a standard multivariate Bayesian linear regression models. The full conditionals of the atoms $\{\Phi_{0h}\}$ in the stick-breaking prior (9.2) are given in the blocked Gibbs sampling of Ishwaran and James (2001).

The code has been implemented in `C++` and linked to `Python` via `pybind11` (Jakob et al., 2017).

9.4 Simulation study

We now present a simulation study to compare the performance of the proposed approach in (9.2)-(9.3) versus a similar model but assuming the Φ_i's to be generated as independent and identically distributed from a Dirichlet Process (DP, Ferguson, 1973) which is arguably the most popular Bayesian nonparametric prior (see, e.g. Müller et al., 2015).

We consider three different simulation scenarios. In scenarios (I) and (II) the responses are simulated from (9.1), while in scenario (III) we simulate each ε_{itj} in $\boldsymbol{\varepsilon}_{it} = (\varepsilon_{it1}, \ldots, \varepsilon_{itk})$ from a student-t distribution with mean 0 and 5 degrees of freedom, so that our model is then misspecified. Of course, other kind of misspecifications are possible, for instance, we could generate data from an autoregressive process with a larger lag, but this would lead to much poorer results for any model with our likelihood. Summing up, though scenario (I) and (II) simulate points from the same conditional distribution we assume for our model, they differ (see below) by the *true* number of clusters and the simulated covariates (well separated only in scenario (I)). Scenarios (I) and (II) were designed to (*i*) check the algorithm and the code, (*ii*) check the ability of the model to detect the right number of clusters, while scenario (III) tests the ability of the models (ours and the competitor) when the model is misspecified. The two models are tested computing out-of-sample and in-sample prediction errors, as well as adjusted Rand Index between the estimated and true partition of the sample.

For all scenarios, we simulate $N = 300$ independent trajectories, namely $\boldsymbol{y}_i = (\boldsymbol{y}_{i1}, \ldots, \boldsymbol{y}_{iT_i})$, with $T_i = 10$ for all i, assuming each $\boldsymbol{y}_{it}$ to be a three-dimensional vector (i.e., $k = 3$).

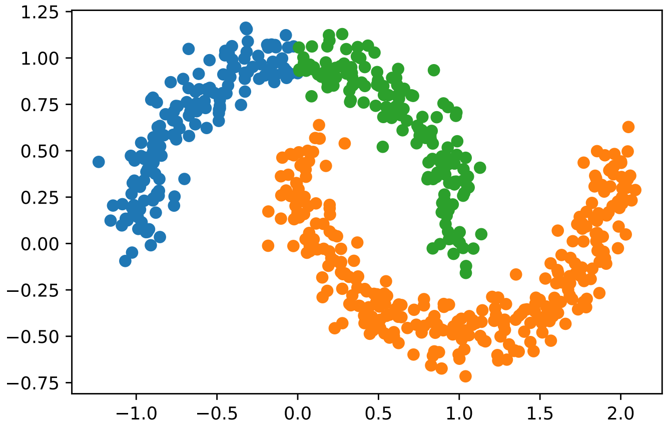


Figure 9.4.1: Fixed time covariates for scenario (II)

Moreover, we always set $B = \mathbf{0}$, $\Gamma = 0$ in the data generating process. In all the scenarios, for each item i, we simulate Φ_i from a discrete mixture, $\Phi_i \overset{\text{iid}}{\sim} \sum_{j=1}^{3} \pi_j \delta_{\bar{\phi}_j}$, where the $\bar{\phi}_j$'s are given in (9.7) (see here below). Then, conditionally to Φ_i we generate the time-homogeneous covariate vector $\boldsymbol{z}_i$. In scenarios (I) and (III) the weights (π_1, π_2, π_3) are set equal to $(0.5, 0.5, 0)$, $\boldsymbol{z}_i \mid \Phi_i = \bar{\phi}_1 \sim \mathcal{N}_2((-3,-3), I_2)$ and $\boldsymbol{z}_i \mid \Phi_i = \bar{\phi}_2 \sim \mathcal{N}_2((3,3), I_2)$

$$\bar{\phi}_1 = \begin{bmatrix} 1.1, 0.0, 0.0 \\ 0.0, 1.1, 0.0 \\ 0.0, 0.0, 1.0 \end{bmatrix}, \quad \bar{\phi}_2 = \begin{bmatrix} 1.1, -0, 1, 0.0 \\ -0.1, 1.1, -0.1 \\ 0.0, 0.0, 0.9 \end{bmatrix}, \quad \bar{\phi}_3 = \begin{bmatrix} 0.9, -0, 1, 0.0 \\ -0.1, 1.1, -0.1 \\ -0.1, 0.0, 1.5 \end{bmatrix}, \tag{9.7}$$

while in scenario (II) the weights are $(0.25, 0.25, 0.5)$ and the simulated time-homogeneous covariates are reported in Figure 9.4.1. Observe that while in scenario (I) and (III) the covariates in the different clusters are clearly separable, this is no longer the case in scenario (II). Finally, in scenarios (I) and (II) we fix $\Sigma = 0.25I$ in (9.1), while in scenario (III) the error terms are generated from a student-t distribution as previously explained.

In the simulations, we set the hyperparameters in (9.6) as follows: $\Phi_{000} = 0$, $\lambda = 0.1$, $V_{00} = I_9$, $\tau_0 = 11$. Moreover, we fix $\Sigma_0 = I_3/\nu$ and $\nu = 5$ (see (9.4)), so that Σ^{-1} has prior mean equal to I_3. For our model, we further assume $\mu_\alpha = \mathbf{0}$, $\Sigma_\alpha = I_9$, $\Sigma_B = I_2$ and $\Sigma_\Gamma = I_{18}$; see (9.3)-(9.4). For the alternative Dirichlet process prior on the Φ_i's, we consider the truncated stick-breaking approximation (Ishwaran and James, 2001), with total mass parameter equal to 1. For both priors the number of atoms H is set equal to 25.

We assess predictive performance of both models through out-of-sample prediction and l-steps ahead in-sample prediction for observed samples. In the first case (later referred to as OOS), for all scenarios, we generate a new test set of size 300 following the same data generating process outlined above, while in the second experiment (INS) we randomly pick 100 of the 300 trajectories generated and 'truncate' them at $T = 5$. In the first setting, the goal is to predict the whole time trajectory given responses at time 1.

We expect our model under prior (9.2)-(9.3) to perform much better than when Φ_i are iid from the Dirichlet process, since our model can assign data to clusters based on their time-homogeneous covariates, while the DP prior cannot. In the second setting, the goal is to predict $l = 5$ steps in the future, i.e. predict $\boldsymbol{y}_{i6}, \ldots, \boldsymbol{y}_{i10}$, for the 100 truncated trajectories. Observe that in this case, we condition on the cluster membership inferred through the MCMC simulation, so that the fixed time covariates are not used to assign trajectories to clusters. As such, we expect the DP prior to have better predictive performance than our model since the number of parameters is considerably smaller compared

	Scenario (I)		Scenario (II)		Scenario (III)	
	LSB	DP	LSB	DP	LSB	DP
OOS	7.5 ± 6.6	65.41 ± 58.8	5.8 ± 3.3	41.9 ± 23.5	91 ± 141	623 ± 1604
INS	3.45 ± 3.12	3.43 ± 3.02	3.9 ± 8.4	4.0 ± 8.5	60.7 ± 114	60.4 ± 113
ARI	1.0	1.0	0.98	0.9	1.0	1.0

Table 9.4.1: Simulated dataset: out-of-sample (OOS) and in sample (INS) mean squared prediction errors and Adjusted Rand Index (ARI) for our model (LSB) and the Dirichlet Process prior for Φ_i's parameters (DP).

to our model. Finally, we also consider the quality of the estimated random partition of the subjects/datapoints, by computing the Adjusted Rand Index (Hubert and Arabie, 1985) between the point estimate of the partition, obtained by minimizing the Binder loss function with equal missclassification cost (see, e.g., Lau and Green, 2007), based on the MCMC samples and the true partition given by the data generating process.

Goodness-of-fit indices shown in Table 9.4.1 confirm our expectations for the out-of-sample testing setting(OOS), that is the proposed approach (denoted in the table as LSB, *logit stick-breaking*) outperforms the DP prior in terms of mean squared prediction. It is clear that for the in-sample predictions both models performs similarly. Note that our model has a slightly better accuracy in terms of clustering for setting (II). This is likely due to the fact that clustering estimation is based also on covariate information and not only on response patterns. The posterior distribution from the DP model favours a larger number of clusters to better approximate the heavy tails of the error's distribution.

Figure 9.4.2 shows posterior predictive distributions for both priors under comparison, considering both out-of-sample and in-sample predictions for scenario (I). We can see that in the OOS case the credible bands for the DP prior are very wide, while those under our model are much narrower. Further, in the INS case, both models display better predictive performance and narrower credible bands.

Finally, we simulate a new dataset under scenario (I), but fixing $\bar{\phi}_2 = \mathbf{0}$ so that the corresponding trajectories are well separated; we note that there is no substantial difference in posterior inference. Figure 9.4.3 reports kernel density estimates from the MCMC sample of the predictive distribution of Φ_i for scenario (I), for three new observations with time-homogeneous covariates equal to $(-3, 0)$, $(3, 0)$ (which coincide with the means of the first and the second group of simulated data) and $(0, 0)$ respectively. Note that the predictive distribution associated to covariate vector equal to $(0, 0)$ (reported in green in Figure 9.4.3) is bimodal, giving almost equal mass to values near $\bar{\phi}_1$ and $\bar{\phi}_2$.

This simulation study shows that the proposed models based on a covariate dependent prior outperforms non-dependent alternative prior in terms of prediction. Moreover, also in terms of clustering structure recovery, the covariate dependent prior gives better estimates in case of heavy tail data.

9.5 Child Growth data

In this section we present posterior results for the Child Growth dataset, detailing prior specification (Section 9.5.1) and inference (Section 9.5.2). In the latter case we include a comparison between our prior and the linear-DDP prior, and also with a parametric counterpart of our model. As a reminder, the dataset contains information on $N = 766$ children with $k = 2$ responses, height and weight of the children over time; see Section 9.2.1 for more details.

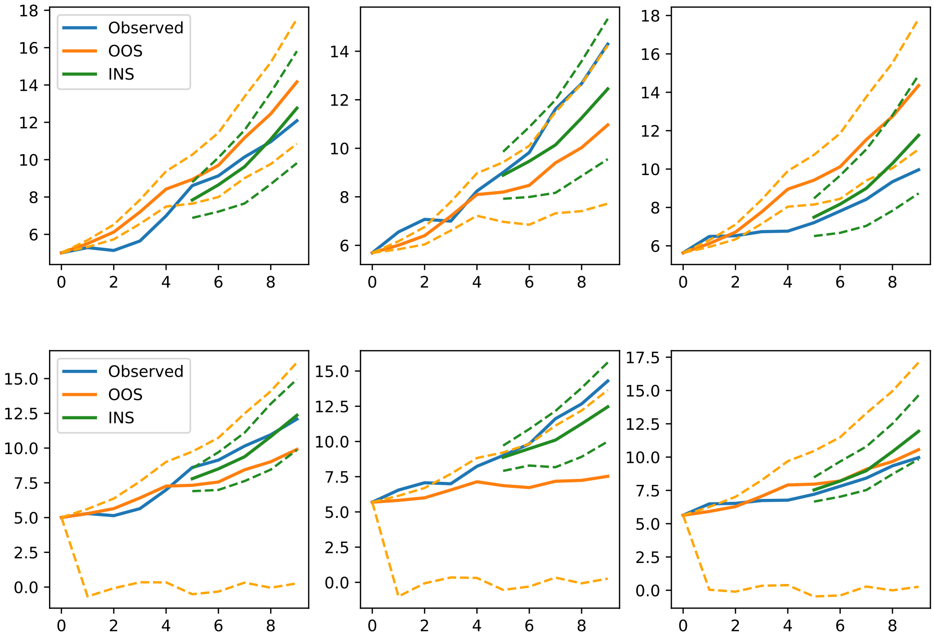


Figure 9.4.2: Posterior predictive distributions for both priors under comparison, considering both out-of-sample and in-sample predictions for scenario (I). We show predictive density estimates and credible intervals using our model (top row) and the DP prior (bottom row) for a new subject i. In each panel, the solid blue lines denote the observed trajectory. The OOS prediction (i.e. given $\boldsymbol{z}_i$ and $\boldsymbol{y}_{i1}$) is shown in orange, while the INS prediction (i.e. given $\boldsymbol{y}_{i5}$ and the cluster label c_i) is shown in green. Solid lines correspond to the median for each time while dashed lines correspond to 95% credible bands of the predictive distributions.

9.5.1 Prior elicitation

Given the complexity of the model and the high-dimensionality of the dataset, prior elicitation needs to be carefully considered. Preliminary analysis shows that when the variances of the α_h's (see (9.3)) or of the atoms Φ_{0h}'s (see (9.5)) in the logit stick-breaking are large, then all the observations tend to be assigned to the same component. Moreover, the missing data simulation step has a strong impact on posterior inference. In particular, when using the vague prior described above, in the initial iterations of the MCMC algorithm, typically large missing values were imputed (e.g. 10^5) since both Σ and $\{\Phi_{0h}\}$ would take on unusually large values. Consequently, sampled values for all the other parameters are affected, leading to a poor fit. Hence the use of an uninformative prior is not advisable, causing poor mixing and slow convergence of the chain. Moreover, this is a common situation in complex hierarchical models when non-informative priors are adopted in lower levels.

As such, we opt for informative priors. To set the hyperparameters in the hierarchical marginal prior in (9.5)-(9.6), we first obtain the maximum likelihood estimator from a vector autoregressive model:

$$\boldsymbol{y}_{it} \,|\, \boldsymbol{y}_{it-1} \sim \mathcal{N}(\Phi \boldsymbol{y}_{it-1}, \Sigma), \qquad t = 1, \dots T-1,\, i = 1, \dots, N \tag{9.8}$$

which corresponds to (9.1) when B and Γ are set to zero (their prior expected value) and $H = 1$. We fit (9.8) using only subjects with no missing responses. Let $\widehat{\Phi}$, $\widehat{\Sigma}$ denote the maximum likelihood estimator for Φ and Σ respectively. We fix $\Phi_{000} = \widehat{\Phi}$, $\lambda = 1$, and

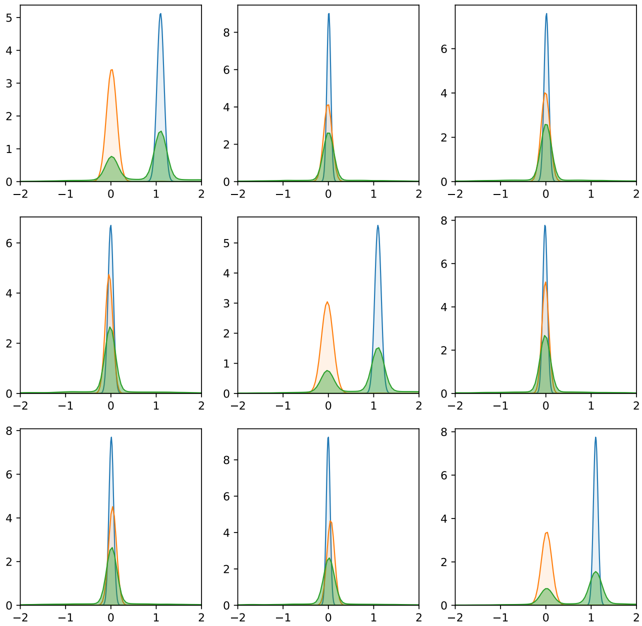


Figure 9.4.3: Predictive distributions of Φ_i^{new} corresponding to three subjects with fixed-time covariates equal to $(-3, 0)$ (blue) $(3, 0)$ (orange) and $(0, 0)$ (green), respectively.

select (V_{00}, τ) in (9.6) so that $\mathbb{E}[V_0] = I$ and $\mathrm{Var}[\{V_0\}_{ii}] = 1.5$. Similarly, we fix Σ_0 and ν in (9.4) so that $\mathbb{E}[\Sigma] = \widehat{\Sigma}$ and $\mathrm{Var}[\{\Sigma_{ii}\}] = 10$.

The variance hyperparameter Σ_α in (9.3) also has an important effect on posterior inference. To set this quantity, we look at the prior distribution of the number of clusters (i.e. *occupied components*) and of the size of the largest cluster. To this end, we perform Monte Carlo simulations. Specifically, we fix the number of components H in the stick-breaking prior equal to 50, set $\Sigma_\alpha = \sigma_\alpha^2 I$, and simulate $\alpha_1, \ldots, \alpha_{H-1}$ from (9.3) with $\mu_\alpha = \mathbf{0}$. Then, for each of the $N = 766$ subjects, we compute the associated weights $\boldsymbol{w}(\boldsymbol{z}_i)$ from the logit stick-breaking process, using observed covariates $\boldsymbol{z}_i$, and allocate each subject to one of the H components with probability given by the weights $\boldsymbol{w}(\boldsymbol{z}_i)$. The above procedure is repeated independently for $M = 10,000$ iterations and we record the number of clusters and the size of the largest cluster. Figure 9.5.1 shows the distributions obtained from the Monte Carlo simulation. As σ_α^2 increases, the number of clusters shrinks to 1 and the size of the largest cluster increases accordingly. Hence, we fix $\sigma_\alpha^2 = 5$ so that a priori we should expect approximately $4 - 7$ clusters. Finally, we assume $\mu_\alpha = \mathbf{0}$, $\Sigma_B = I_2$ and $\Sigma_\Gamma = I_{18}$ (see (9.4)); recall that all continuous covariates are standardized.

9.5.2 Posterior inference results

We apply the model described in Section 9.3 to the Child growth dataset with hyperparameters set as in Section 9.5.1. Recall (Section 9.2.2) that the model includes $p = 1$ time-dependent covariate (that is $\sqrt{t}$) and a $q = 14$-dimensional design matrix for time-homogeneous covariates (including intercepts, interactions and dummy variables to represent categorical covariates). We run the MCMC algorithm for 100,000 iterations, discard-

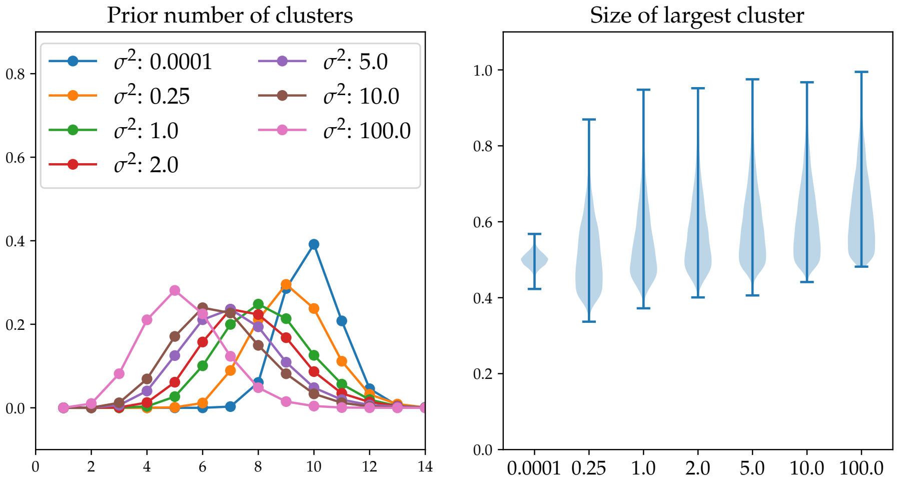


Figure 9.5.1: Prior distribution of the number of clusters (left panel) and of the size of the largest cluster as percentage of the whole dataset (right panel), for different values of σ_α.

ing the first 50,000 as burn-in and thinning every 10 iterations, obtaining a final sample size of 5,000 iterations.

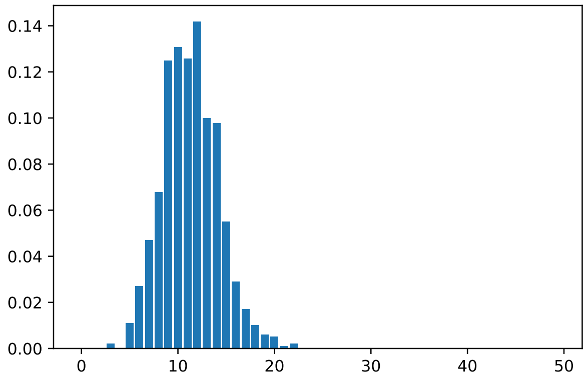


Figure 9.5.2: Child Growth dataset: posterior distribution of the number of clusters.

Figure 9.5.2 shows the posterior distribution of the number of clusters, i.e. of *occupied* parametric components, that is clearly centered around 10-12 clusters. However, interpreting these as the 'number of distinct profiles' in the $\boldsymbol{y}$'s may be misleading. Recall that we have specified a covariate-dependent prior for the random partition of patients. Indeed, some clusters can be essentially identical when looking at the response trajectories but different when looking at the covariates. As a point estimate of the latent partition, we choose the one that minimises the Binder loss function under equal misspecification costs (Binder, 1978). The estimated partition consists of seven clusters, of which only four contain at least 15 observations. In Figure 9.5.3 we display the response trajectories clustered according to the estimated partition. Note that the fourth cluster (bottom row) consists of subjects with at most three visits, except for one single subject with four visits. For this reason, we do not discuss this cluster. Figure 9.5.3 shows the time trajectories for patients' height (first column), weight (second column) and BMI. The third row in Figure 9.5.3 shows that this cluster contains children with lower weight, and consequently lower BMI than the other two clusters.

As already mentioned, the main three clusters could differ either in the responses or in the covariates (or both). To better understand what discriminates the three main clusters, we perform homogeneity tests for the equality in distribution of both responses and covariates in the different clusters. The results should be considered as a descriptive tool. In particular, for the responses we consider the data on both height and weight at each visit separately and test the equality of the distributions for each pair of clusters. For each of the covariates, we test the equality of their distributions in each possible pair of clusters. For the response variables and continuous covariates, we employ the Kolmogorov-Smirnov (KS) test for equality in distribution and the Pearson's chi- squared test of homogeneity for the categorical covariates. Table 9.5.1 reports the p-values associated to the KS test for the responses, while Figure 9.5.4 shows the cluster specific empirical distribution of the covariates. From Table 9.5.1 and Figure 9.5.4, it is clear that clusters 2 and 3 (second and third rows in Figure 9.5.3, respectively) are similar in terms of both responses at each time point. However, Figure 9.5.4 (bottom row) suggests that the three main clusters cannot be *explained* only in terms of *ethnicity*, even though cluster 3 contains almost exclusively Chinese children.

	Height			Weight		
Clusters	(1, 2)	(1, 3)	(2, 3)	(1, 2)	(1, 3)	(2, 3)
$t = 1$	**0.023**	**0.000**	**0.025**	**0.002**	**0.296**	0.606
$t = 2$	**0.000**	**0.023**	0.999	**0.000**	**0.000**	0.785
$t = 3$	**0.000**	**0.003**	0.797	**0.000**	**0.013**	0.815
$t = 4$	**0.000**	**0.000**	**0.000**	**0.000**	0.253	0.620
$t = 5$	**0.046**	**0.004**	**0.044**	**0.000**	0.197	0.386
$t = 6$	**0.000**	0.051	0.701	**0.000**	0.241	0.254
$t = 7$	**0.003**	0.113	0.878	**0.000**	0.431	0.375
$t = 8$	**0.000**	0.072	0.733	**0.000**	0.210	0.718
$t = 9$	**0.000**	0.106	0.984	**0.000**	0.196	0.715
$t = 10$	**0.000**	0.112	0.869	**0.000**	0.341	0.717
$t = 11$	**0.000**	0.213	0.726	**0.000**	0.244	0.854
$t = 12$	**0.000**	0.165	0.877	**0.000**	0.125	0.932
$t = 13$	**0.000**	0.179	0.993	**0.000**	**0.042**	0.811

Table 9.5.1: P-values of the homogeneity tests for the equality in distribution at every visit for each pair of clusters, considering height and weight. Bold numbers correspond to p-values lower than 5%

Next we consider the two parameters in B, i.e. the regression parameters for the square root of time t for the two responses; see (9.1). The posterior means are $5.55, 0.96$, respectively, with marginal standard deviations $0.02, 0.01$, thus indicating a non-negligible growth trend for both height and weight, as expected. Figure 9.5.5 displays posterior credible intervals for all the parameters in Γ defined in (9.1), that is the regression coefficients corresponding to the time-homogeneous covariates. The reference group for the categorical covariates has been set such that the baseline level is for a Chinese female child; see Section 9.2.2. Covariates such as *OGTT 2h*, *ppBMI*, the interaction between education and age, ethnicity (Malay) and the interaction between sex and ethnicity have the strongest effects on height. On the other hand, *parity*, *OGTT 2h*, *ppBMI*, the interaction between education and age (but only the second level of education) and the interaction between sex and ethnicity have a strong association with weight. It is clear from Figure 9.5.5 that most of the posterior mass for the marginal distribution of *ethnicity* is concentrated on

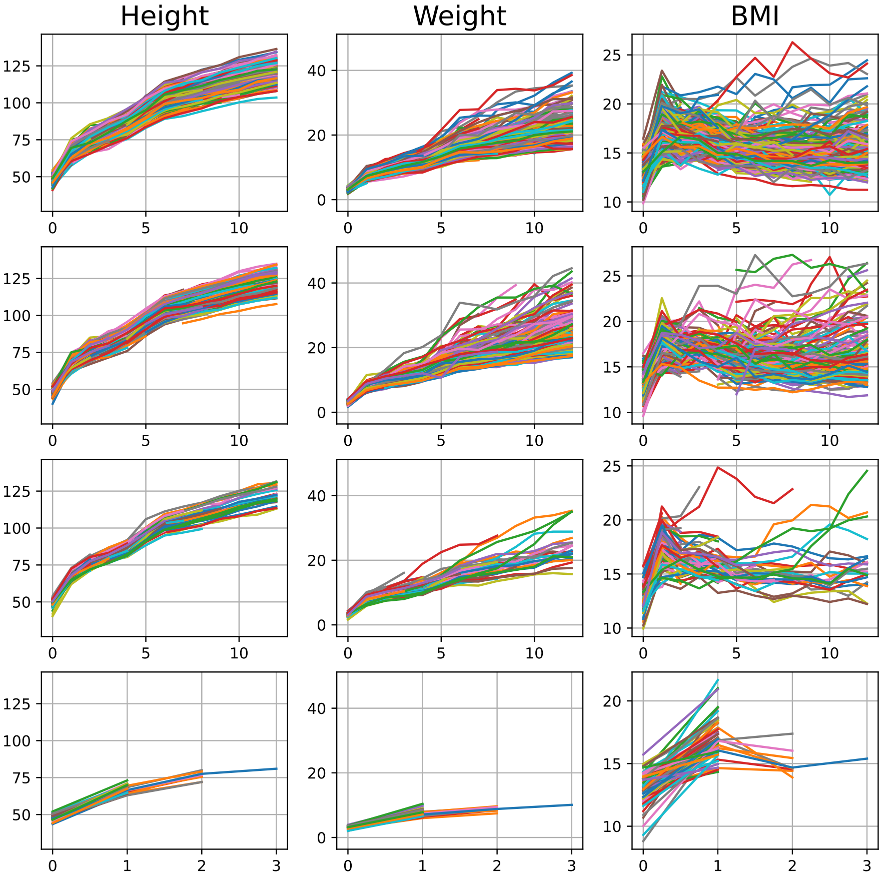
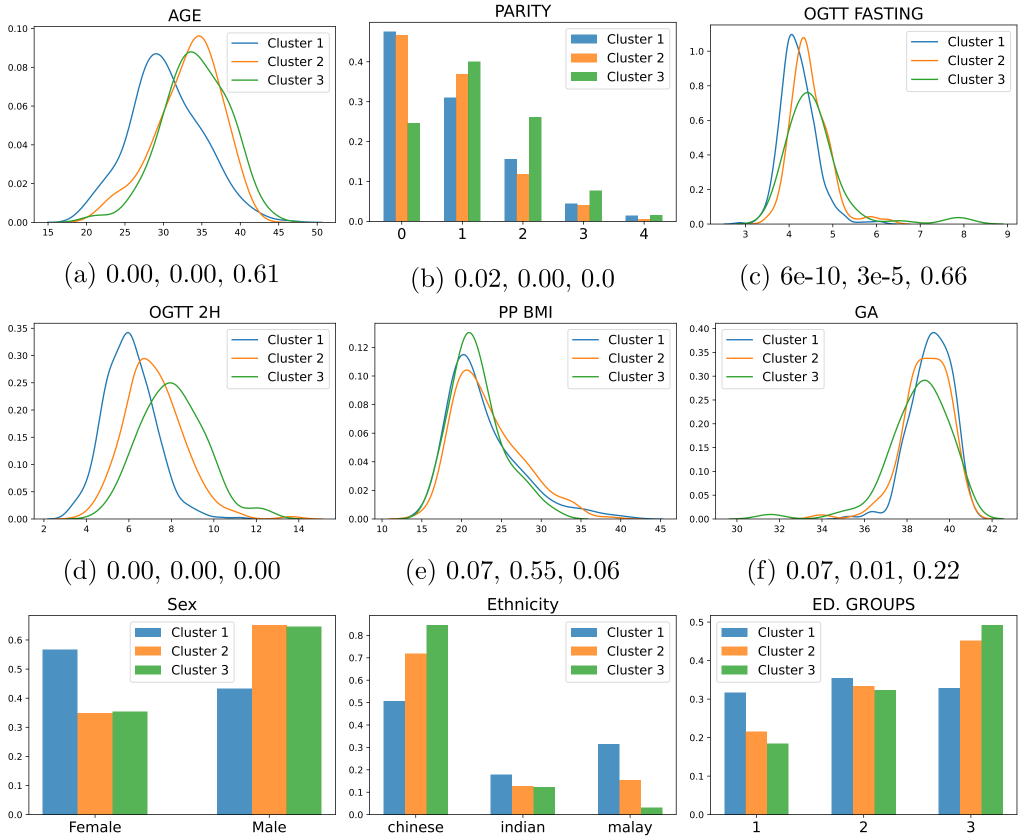


Figure 9.5.3: Subject trajectories of height (first column), weight (second column) and BMI (third column) by estimated cluster (by row). The figure reports only the four largest clusters out of the seven estimated.

positive values. Correcting for the autoregressive effect, we see that *ethnicity* might impact obesity as Indian and Malay children are characterised by a larger posterior expected weight, combined in some cases with a lower posterior expected height. Moreover, also correcting for the autoregressive effect, our analysis shows that the posterior expected height of a Chinese male child is larger than the reference (Chinese female child). Similar comments can be made, for instance, regarding Indian male children being smaller than Indian female children, and so on.

Mother's age and gestational age do not have a strong effect on the child's height and weight, though this might be due to the fact that these variables are associated with ethnicity; see Figure 9.2.2. It is known from the literature that increasing parity is associated with increasing neonatal adiposity in Asian and Western populations (see Tint et al., 2016); this is confirmed by the marginal posterior distribution of the parameter corresponding to the effect of *parity* on weight in Figure 9.5.5.

The z_i covariates play also a key role in defining the stick-breaking prior as seen from (9.3). To assess if the proposed covariate-driven stick-breaking prior provides significant



(a) 0.00, 0.00, 0.61

(b) 0.02, 0.00, 0.0

(c) 6e-10, 3e-5, 0.66

(d) 0.00, 0.00, 0.00

(e) 0.07, 0.55, 0.06

(f) 0.07, 0.01, 0.22

(g) 0.00, 0.00, 0.98

(h) 0.00, 0.00, 0.00

(i) 0.00, 0.00, 0.40

Figure 9.5.4: Empirical distribution of the covariates in each cluster. The three numbers below each plot represent the p-values for the homogeneity tests for covariates in clusters (1, 2), (1, 3) and (2, 3), respectively.

advantages over more standard models, we compare it with three possible competitors. The first one is the parametric version of our model obtained by setting $H = 1$. The second model assumes a truncated Dirichlet process as a prior for Φ_i's, with $H = 50$, similarly to what is done in Section 9.4. Moreover, as third competitor prior, we assume that the Φ_i's take into account information from the time-homogeneous covariates through the atoms Φ_{0h}'s. Specifically, the prior for Φ is specified as in (9.2), but for each $h = 1, \dots, H$ we define a matrix $\Omega_h \in \mathbb{R}^{k^2 \times q}$ and we let $vec(\Phi_{0h}(\boldsymbol{z}_i)) =: \varphi_{0h}(\boldsymbol{z}_i) = \Omega_h \boldsymbol{z}_i$. The weights $\boldsymbol{w}$ in (9.2) do not depend on the value of $\boldsymbol{z}_i$ (i.e., $w_h(\boldsymbol{z}_i) = w_h$) and follow a truncated Dirichlet process prior with $H = 50$. This model can be seen as a finite dimensional approximation of the Linear-DPP in De Iorio et al. (2004).

For all the models, we match the prior for B, Γ, Σ and, when possible also H and the marginal prior distribution of Φ_{0h}. For the Linear-DPP we assume that the vectorization of the Ω_h's are independent and identically distributed multivariate Gaussian random variables with mean zero and identity covariance matrix. Since the full conditional distribution of the Ω_h's in the case of the Linear-DPP prior does not belong to a known parametric family, we update them via an adaptive Metropolis Hastings (Andrieu and Thoms, 2008) step.

The different models are compared using widely applicable information criterion (WAIC, Watanabe, 2013). Higher values of WAIC correspond to better predictive performances. We marginalize the missing values from the predictive distribution of the response trajec-

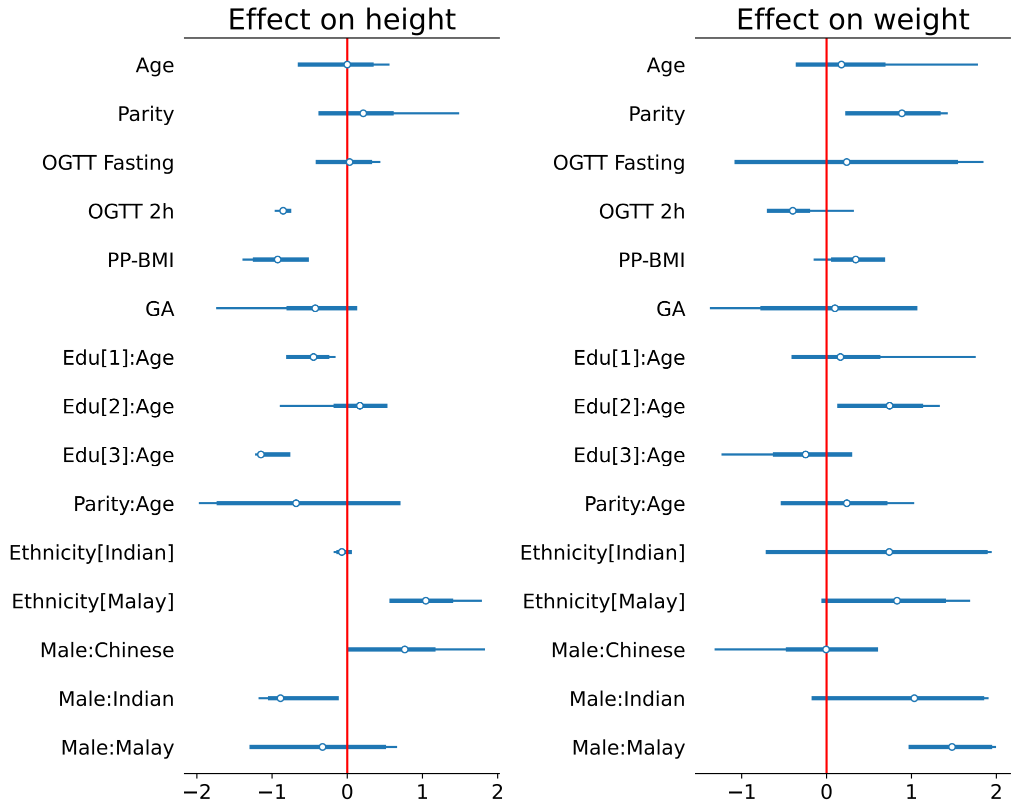


Figure 9.5.5: Posterior credible intervals of the regression coefficients in Γ for the height (left plot) and weight of the children (right plot). Thin lines correspond to 95% credible intervals, while thick lines to 80% credible intervals.

tory and consider just the marginal predictive distribution for the non-missing values. We found that WAIC is equal to -3.4×10^6 for the Linear-DDP, -6.7×10^5 for the parametric model, -3.9×10^5 for the DP model and -3.4×10^5 for our model, confirming that our model performs better than the competitors. Moreover, we report that the MCMC algorithm for the Linear-DDP requires a much larger number of burn-in iterations (10^5 vs. 10^4) than the other models to reach satisfactory convergence, and that the expected number of cluster a posteriori in the Linear-DPP is around 42. It is then clear that (i) assuming linear dependence of the fixed-time covariates in the autoregressive parameters matrices Φ_i does not give good predictive fit (or at least not better than our model), and that (ii) adding covariate information in the stick-breaking prior improves the prediction performance.

9.6 Summary

The aim of this manuscript is to cluster children according to obesity growth patterns. Obesity is an epidemic, increasingly affecting children. Overweight or obesity in childhood may be critical as they often persist into adulthood due to both physiological and behavioral factors.

Motivation for our study stems from a child growth dataset. To analyze these data we developed a Bayesian nonparametric VAR joint model for height and weight profiles for these children. One key aspect behind the modeling choice was to cluster the corresponding joint time-evolving profiles using the available covariate information. The model features a logit stick-breaking construction that can accommodate covariate dependence in the mixture weights. This allows us to relate certain baseline conditions of these children, such as sex or ethnicity, to obesity patterns. Ethnic differences in obesity are of interest as they could be due to genetic factors, dietary intake, cultural or socioeconomic factors. The analysis allowed us to identify important clusters of children that are characterized by differences in the trajectories or in the covariates or both.

Posterior inference was carried out by means of an efficient posterior simulation that exploits recently developed results on logit stick-breaking priors, which facilitates postulating covariate dependence in the mixture weights. For this implementation we chose to fix a sufficiently large number of components from which we focused on the number of these that were actually occupied (we referred to these as *clusters*). The results obtained were compared against competitor models, and we found that our approach provides superior performance as measured by standard quantities such as the WAIC.

An interesting characteristic of our model is that, though it clusters the obesity patterns of the children in the study, when we aim at interpreting the estimated clusters in terms of 'number of distinct profiles' in the responses, we should also take into account that the prior we assume for the random partition of the sample subjects is covariate-dependent. In fact, some of the estimated clusters are similar when looking at the response trajectories but different in terms of the associated covariates. We consider this aspect as an advantage of our model (and all models with covariate-dependent prior for the random partition), that allows for greater flexibility for clustering, rather then an inadequacy.

Given the goal of clustering the children obesity patterns using extra information on children and their mothers, we have made specific choices for the different parts of the models, i.e., AR structure in the likelihood, the mean temporal trend, the interactions in the linear regression term, the BNP prior for clustering. In particular, the logit stick-breaking Bayesian nonparametric prior exploits the potential and interpretability of logistic regression (wrt linear dependent Dirichlet process or probit stick-breaking priors) combined with recent computational schemes by Rigon and Durante (2021).

Finally, we mention that in building the proposed model, several sensible alternative options could have been adopted. Some of them were discussed throughout the manuscript. Nevertheless, the preliminary and exploratory analysis, together with the predictive checks carried out confirm that our modeling choices are reasonable for the data under consideration.

Appendix

9.A Further plots

We show the scatterplots of the responses (height and weight) at time $t = 0, 1, 2$ versus all continuous covariates at the baseline of the dataset on obesity for Children in Singapore. When the covariate we consider is discrete, scatterplots are replaced by boxplots. The left column of Figure 9.A.1 reports scatterplots or boxplots of the height at time $t = 0$, while the left column of Figure 9.A.2 reports similar plots for the weight at time $t = 0$. The central and right columns of Figures 9.A.1 and 9.A.2 display the same plots of the responses at time $t = 1$ and 2.

9.B Details on the Gibbs sampler

Posterior inference for our logit stick-breaking model (9.1)-(9.6) is carried out using a Gibbs sampler algorithm, with full conditionals outlined below. The joint distribution of data and parameters is described here

$$\mathcal{L}(\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_N, B, \Gamma, \Sigma, \Phi_1, \dots, \Phi_N) = \prod_{i=1}^{N} \mathcal{L}(Y_{i1}, \dots, Y_{iT_i} | \boldsymbol{b}, \boldsymbol{\gamma}, \Sigma, \Phi_1, \dots, \Phi_n)$$
$$\times \pi(\boldsymbol{b}) \times \pi(\boldsymbol{\gamma}) \times \pi(\Sigma) \times \pi(\Phi_1, \dots, \Phi_N | \boldsymbol{z}_1, \dots, \boldsymbol{z}_N)$$

In what follows, 'rest' refers to to the data and all parameters except for the one to the left of '$|$'. Moreover we adopt the matrix notation or the vector one for all parameters interchangeably.

As in Ishwaran and James (2001), to sample from the stick-breaking prior on Φ_i, as it is standard, we use cluster indicator latent variables, that will be indicated by G_i.

1. The full-conditional for the parameters $\boldsymbol{b} = \mathrm{vec}(B)$ can be obtained by noticing that using the following change of variable

   $$\boldsymbol{y_{it}} - \Phi_i \boldsymbol{y_{it-1}} - \Gamma \boldsymbol{z_i} = B \boldsymbol{x_{it}} + \boldsymbol{\epsilon_{it}}$$

   we recover the standard expression of Bayesian multivariate linear regression, let $\boldsymbol{w_{it}} = \boldsymbol{y_{it}} - \Phi_i \boldsymbol{y_{it-1}} - \Gamma \boldsymbol{z_i}$. We have:

   $$\boldsymbol{w_{it}} = \boldsymbol{x_{it}^T} B^T + \boldsymbol{\epsilon_{it}}.$$

   Using standard techniques, calling

   $$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_{1,1} \\ \vdots \\ \boldsymbol{w}_{i,T_1} \\ \vdots \\ \boldsymbol{w}_{N,1} \\ \vdots \\ \boldsymbol{w}_{N,T_N} \end{bmatrix} \qquad X = \begin{bmatrix} \boldsymbol{x}_{1,1} \\ \vdots \\ \boldsymbol{x}_{1,T_1} \\ \vdots \\ \boldsymbol{x}_{N,1} \\ \vdots \\ \boldsymbol{x}_{N,T_N} \end{bmatrix}$$

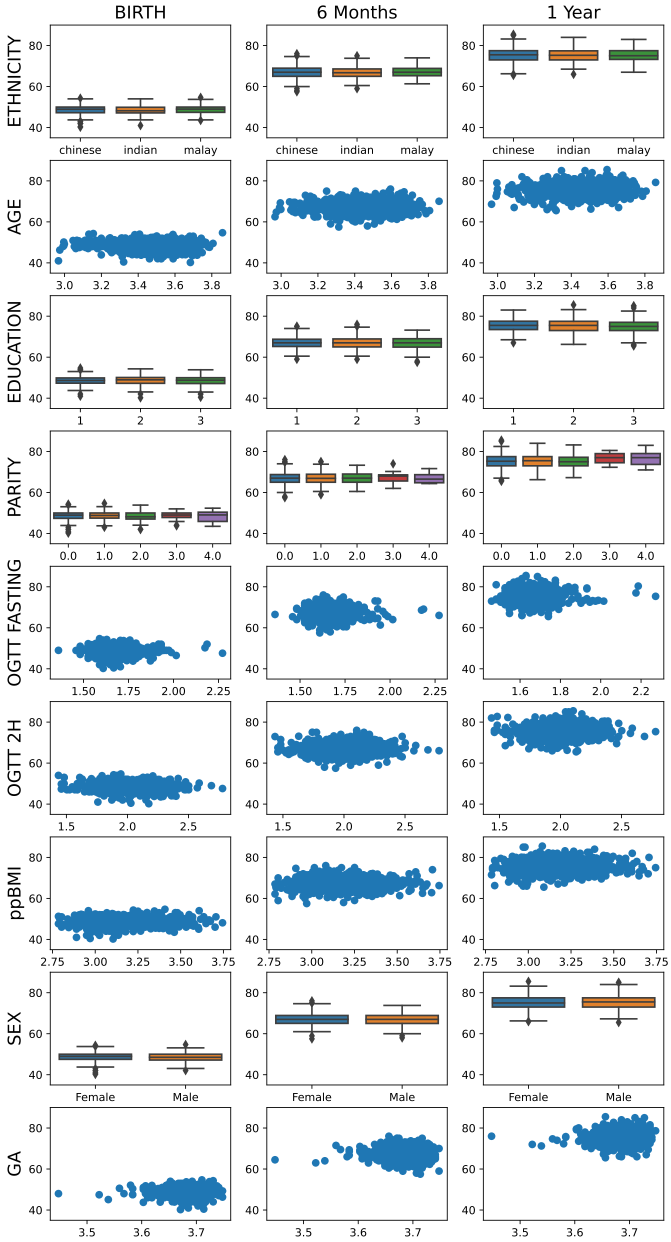


Figure 9.A.1: Scatterplots of covariates against the height at birth (left column), at six months of age (center) and at one year of age (right column).

We can write the system in vector form as:

$$\boldsymbol{W} = XB^T + \boldsymbol{E},$$

where W, E are $[\sum_{i=1}^{N} T_i \times k]$ matrices and X is $[\sum_{i=1}^{N} T_i \times p]$. By standard multivariate regression theory we have that

$$\begin{aligned} \boldsymbol{b}|X, W, \Sigma &\sim \mathcal{N}\left(\widetilde{\mu_b}, \widetilde{\Sigma_b}\right) \\ \mu_b &= (\Sigma^{-1} \otimes X^T X + \Sigma_b^{-1})^{-1} \left((\Sigma^{-1} \otimes X^T X)\hat{\beta} + \Sigma_b^{-1}\widetilde{\beta}_0\right) \\ \widetilde{\Sigma_b} &= \Sigma^{-1} \otimes X^T X + \Sigma_b^{-1}, \end{aligned}$$

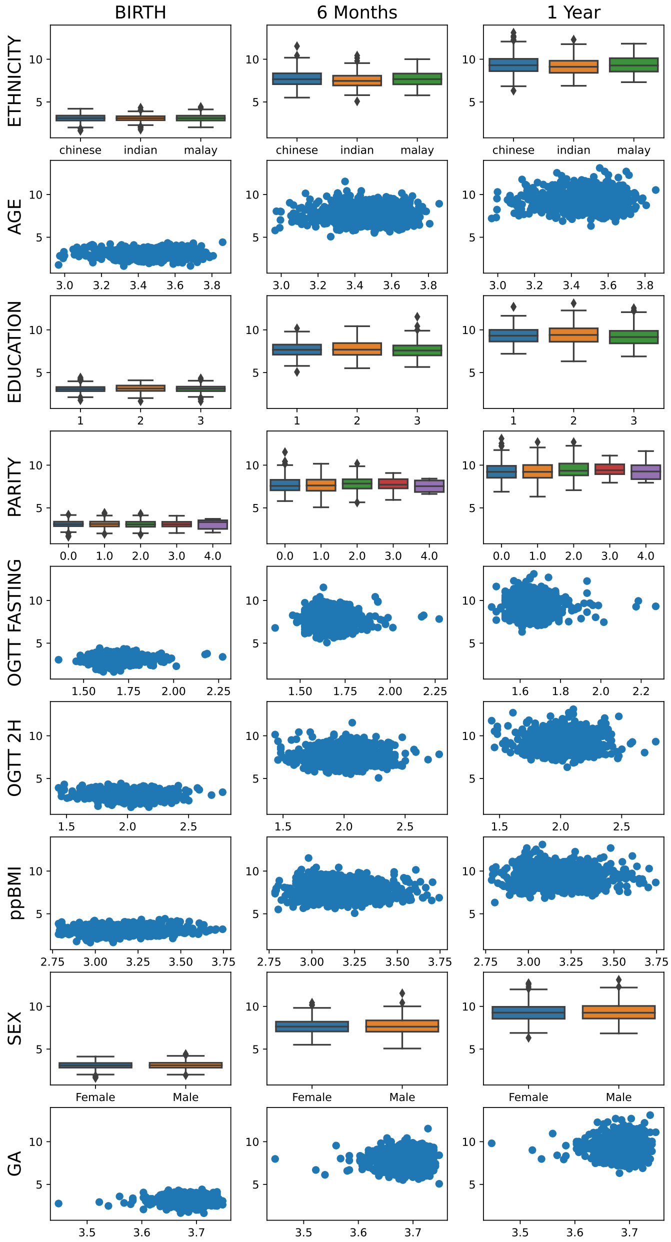


Figure 9.A.2: Scatterplots of covariates against the weight at birth (left column), at six months of age (center) and at one year of age (right column).

where $\hat{\beta}$ is the standard frequentist estimate:

$$\hat{\beta} = (X^T X)^{-1} X^T \boldsymbol{W}.$$

We thus obtain:

$$\mathcal{L}(\boldsymbol{b}|\text{rest}) = \mathcal{N}(\widetilde{\mu_b}, \widetilde{\Sigma_b})$$

2. Analogously to what we did in the previous step, the law of $\boldsymbol{\gamma}$ can be deducted from standard Bayesian multivariate regression theory after a suitable change of variable:

$$\boldsymbol{y}_{it} - \Phi_i \boldsymbol{y}_{it-1} - B\boldsymbol{x}_{it} = \Gamma \boldsymbol{z}_i + \boldsymbol{\epsilon}_{it}$$

We thus recover the same equations as in the previous section.

3. To sample from $\mathcal{L}(\Sigma|\text{rest})$ we analyze the full conditional (to simplify the notation we impose $\boldsymbol{y}_{i0} = \mathbf{0}$ for all i's):

$$\begin{aligned}\mathcal{L}(\Sigma^{-1}|-) &\propto \prod_{i=1}^{N}\prod_{t=1}^{T_i} \frac{1}{det 2\pi\Sigma^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\boldsymbol{\eta}_{it}^{T}\Sigma^{-1}\boldsymbol{\eta}_{it}\right) \\ &\times \frac{det\Sigma_0{}^{\frac{\tau}{2}}}{2^{\frac{\tau k}{2}}\Gamma_k\left(\frac{\tau}{2}\right)} det\Sigma^{-\frac{\tau+k+1}{2}} \exp\left(-\frac{1}{2}tr(\Sigma_0\Sigma^{-1})\right) \\ &\propto \frac{det\Sigma^{-\frac{\tau+k+1}{2}}}{det 2\pi\Sigma^{+\frac{1}{2}\sum_{i=1}^{N}T_i}} \exp\left(-\frac{1}{2}E\right).\end{aligned}$$

where

$$\boldsymbol{\eta}_{it} = \boldsymbol{y}_{it} - \Phi_i\boldsymbol{y}_{it-1} - B\boldsymbol{x}_{it} - \Gamma\boldsymbol{z}_i$$

By using the trace trick, circularity of the trace and linearity of the trace operator we get that

$$E = tr\left(\left(\sum_{i=1}^{N}\sum_{t=1}^{T_i}\boldsymbol{\eta}_{it}\boldsymbol{\eta}_{it}^{T} + \Sigma_0\right)\Sigma^{-1}\right).$$

We can deduce that $\mathcal{L}(\Sigma\,|\,\text{rest}) = IW(\tilde{\nu}, \widetilde{\Sigma_0})$ with parameters

$$\begin{aligned}\tilde{\nu} &= \nu + \sum_{i=1}^{N} T_i \\ \widetilde{\Sigma_0} &= \sum_{i=1}^{N}\sum_{t=1}^{T_i}\boldsymbol{\eta}_{it}\boldsymbol{\eta}_{it}^{T} + \Sigma_0.\end{aligned}$$

4. The component indicator variables are sampled considering the usual change of variables

$$\boldsymbol{w}_{it} = \Phi_i(\boldsymbol{z}_i)\boldsymbol{y}_{it-1} + \boldsymbol{\epsilon}_{it},$$

where $\boldsymbol{w}_{it} = \boldsymbol{y}_{it} - B\boldsymbol{x}_{it} - \Gamma z_i$. We have that:

$$\begin{aligned}P(G_i = h|\text{rest}) &\propto P(G_i = h) f(\boldsymbol{w}_{i1}, \ldots, \boldsymbol{w}_{iT_i}|G_i = h) \qquad (9.9) \\ &\propto P(G_i = h \times f(\boldsymbol{w}_{i1}|G_i = h, \text{rest}) \prod_{t=2}^{T_i} f(\boldsymbol{w}_{it}|\boldsymbol{y}_{it-1}, \text{rest}) \\ &\propto \nu_h(\boldsymbol{z}_i) \prod_{l=1}^{h-1}(1 - \nu_l(\boldsymbol{z}_i)) \times \mathcal{N}(\boldsymbol{w}_{i1}; \boldsymbol{B}x_{i1} + \Gamma z_i, \Sigma) \\ &\quad\times \prod_{t=2}^{T_i}\mathcal{N}(\boldsymbol{w}_{it}; \Phi_{0h}\boldsymbol{y}_{it-1}\boldsymbol{B}x_{it} + \Gamma z_i, \Sigma).\end{aligned}$$

Thus the conditional distribution of G_i is a discrete distribution with weights as in (9.9).

5. For each cluster-specific Φ_{0h} we have that, for the i's such that $G_i = h$:

$$\boldsymbol{y}_{it} = \Phi_{0h}\boldsymbol{y}_{it-1} + Bx_{it} + \Gamma z_i + \epsilon_{it}.$$

Defining:

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_{11} - B\boldsymbol{x}_{11} - \Gamma\boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{y}_{1T_1} - B\boldsymbol{x}_{1T_1} - \Gamma\boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{y}_{N_i1} - B\boldsymbol{x}_{N_i1} - \Gamma\boldsymbol{z}_{N_i} \\ \vdots \\ \boldsymbol{y}_{N_iT_{N_i}} - B\boldsymbol{x}_{N_iT_{N_i}} - \Gamma\boldsymbol{z}_{N_i} \end{bmatrix} \qquad X = \begin{bmatrix} \boldsymbol{y}_{10} \\ \vdots \\ \boldsymbol{y}_{1T_1-1} \\ \vdots \\ \boldsymbol{y}_{N_i1} \\ \vdots \\ \boldsymbol{y}_{N_iT_i-1} \end{bmatrix}$$

where the $\boldsymbol{y}_i$s have been selected such that they belong to cluster h, we have the following Seemingly Unrelated Representation:

$$\boldsymbol{Y} = X\Phi_{0h}^T + E.$$

Thus, we can recover the full conditional for $\varphi_{0h} := \mathrm{vec}\,(\Phi_{0h})$ using standard Bayesian multivariate regression theory. In particular we have that:

$$\begin{aligned} \varphi_{0h}|Y, X, \Sigma &\sim \mathcal{N}(\mu_{0h}, \Sigma_{0h}) \\ \Sigma_{0h} &= \Sigma^{-1} \otimes X^T X + V_0^{-1} \\ \mu_{0h} &= \Sigma_{0h}^{-1}\left((\Sigma^{-1} \otimes X^T X)\widehat{\varphi_{0h}} + V_0^{-1}\varphi_{00}\right), \end{aligned}$$

where $\widehat{\varphi_{0h}} = (X^T X)^{-1} X^T Y$ is the frequentist estimation.

6. Since the update of α_h is independent of the AR model, we can simply refer to Rigon and Durante (2021) where a latent variable ω_{ih} is introduced. Defining $\rho_{ih}|\boldsymbol{z}_i \sim \mathcal{B}(\nu_h(\boldsymbol{z}_i))$, the couple (ω_{ih}, ρ_{ih}) is updated as in Polson et al. (2013) from a Pólya-Gamma distribution.

7. As the joint law does not depend from the parameters Φ_{00}, V_0 except for the prior specification of Φ_{0h}, we can update them using a Normal-Normal-inverse-Wishart scheme as follows:

$$\begin{aligned} \varphi_{0h}|\varphi_{00}, V_0 &\overset{\text{iid}}{\sim} \mathcal{N}(\varphi_{00}, V_0) \\ \varphi_{00}|\varphi_{000}, V_0, \lambda_0 &\sim \mathcal{N}\left(\varphi_{000}, \frac{1}{\lambda_0}V_0\right) \\ V_0|V_{00}, \tau_0 &\sim \mathcal{IW}(V_{00}, \nu_0), \end{aligned}$$

From this we have that:

$$\begin{aligned} \varphi_{00}|V_0, \varphi_{01}, \ldots \varphi_{0H} &\sim \mathcal{N}\left(\frac{H\overline{\varphi_0} + \lambda\varphi_{000}}{H+\lambda}, \frac{1}{H+\lambda}V_0\right) \\ V_0|\varphi_{01}, \ldots \varphi_{0H} &\sim \mathcal{IW}\left(V_P, H + \nu_0\right) \\ V_P &= V_{00} + HS + \frac{H\lambda}{H+\lambda}(\overline{\varphi_0} - \varphi_{000})(\overline{\varphi_0} - \varphi_{000})^T \\ \overline{\varphi_0} &= \frac{1}{H}\sum_{h=1}^{H}\varphi_{0h} \\ S &= \frac{1}{H}\sum_{h=1}^{H}(\varphi_{0h} - \overline{\varphi_0})(\Phi_{0h} - \overline{\varphi_0})^T \end{aligned}$$

An iteration of our Gibbs samples consists in sampling from the full conditionals described in steps 1. through 7. above, iteratively. Moreover, if there are missing responses as in the case of the application, at each iteration, before step 1., we sample the missing responses from their full conditional as described below.

9.B.1 Sampling missing responses

We start by deriving the joint law of the vector $\boldsymbol{y}_i = (\boldsymbol{y}_{i1}, \dots, \boldsymbol{y}_{iT_i})$, given Φ_i, B, Γ and Σ. Consider the simplified VAR model, for a single patient (we drop the index i).

$$\boldsymbol{y}_1 = \epsilon_1, \quad \boldsymbol{y}_t | X_{t-1} = \Phi \boldsymbol{y}_{t-1} + \epsilon_t. \tag{9.10}$$

By expressing the joint law as $\mathcal{L}(\boldsymbol{y}_1, \dots, \boldsymbol{y}_T) = \mathcal{L}(\boldsymbol{y}_1)\mathcal{L}(\boldsymbol{y}_2|\boldsymbol{y}_1)\dots\mathcal{L}(\boldsymbol{y}_T|\boldsymbol{y}_{T-1})$ and through some basic linear algebra, we can derive that the vectorization of $(\boldsymbol{y}_1, \dots, \boldsymbol{y}_T)$ is a jointly normal random vector with zero mean. The precision matrix $\widetilde{\Sigma}^{-1}$ of the normal distribution has a blocked structure made of $T \times T$ blocks, each of which is an $r \times r$ matrix. The (i, j)-th block equals to:

$$\widetilde{\Sigma}^{-1}_{i,j} = \begin{cases} (I + \Phi)^T \Sigma^{-1} (I + \Phi), & \text{if } i = j < T \\ \Sigma^{-1}, & \text{if } i = j = T \\ \Phi^T \Sigma^{-1}, & \text{if } |i - j| = 1 \\ 0 & \text{if } |i - j| > 1 \end{cases} \tag{9.11}$$

Going back to the full model, it is easy to see that with a change of variable $\boldsymbol{y}_{it} \mapsto \boldsymbol{y}_{i,t} - Bx_{i,t} - \Gamma z_i$ we recover the same VAR system in (9.10). Hence, the vectorization of $\boldsymbol{y}_i$ follows a multivariate normal with precision matrix given by (9.11) and mean $\boldsymbol{\mu}$ given by the vectorization of $(B\boldsymbol{x}_{i1} + \Gamma z_i, \dots, B\boldsymbol{x}_{iT_i} + \Gamma z_i)$.

To simulate missing values in $\boldsymbol{y}_i$, we exploit the joint law derived above and the fact that the conditional distributions of entries in a Gaussian random vector are available in close form. In particular, if there are k missing values in $\boldsymbol{y}_i$, we first apply a permutation matrix P to the vectorization of $\boldsymbol{y}_i$ so that the missing entries are the first k (this will in turn change the mean $\boldsymbol{\mu}$ to $P\boldsymbol{\mu}$ and the covariance matrix to $P^T\widetilde{\Sigma}P$). Then, using notation $\boldsymbol{x}^{:k}$ and $\boldsymbol{x}^{k:}$ for the first k elements of vector $\boldsymbol{x}$ and the elements $k+1, \dots$ respectively, and notation $A^{:k,\ell:}$ for a matrix A analogously, where the first index denotes the rows and the second index denotes the columns, we have that:

$$(P\boldsymbol{y}_i)^{:k} \,|\, (P\boldsymbol{y}_i)^{k:} \sim \mathcal{N}_k(\overline{\boldsymbol{\mu}}, \overline{\Sigma}),$$

where

$$\overline{\boldsymbol{\mu}} = [P(B\boldsymbol{x}_i + \Gamma \boldsymbol{z}_i)]^{:k} + [P^T\widetilde{\Sigma}P]^{k:,:k} \left([P^T\widetilde{\Sigma}P)]^{:k,:k} \right)^{-1} (P\boldsymbol{y}_i^{k:} - P(B\boldsymbol{x}_i + \Gamma \boldsymbol{z}_i)]^{k:})$$

and

$$\overline{\Sigma} = [P^T\widetilde{\Sigma}P]^{k:,:k} \left([P^T\widetilde{\Sigma}P)]^{:k,:k} \right)^{-1} [P^T\widetilde{\Sigma}P]^{:k,k:}$$

See Proposition 3.13 in Eaton (1983) for a proof.

10. Distributional data analysis with the Wasserstein metric

The last part of this thesis is dedicated to statistical data analysis when observations are themselves probability measures. This scenario is commonly faced in several important practical applications. For example, when one considers the problem of aggregating (probabilistic) expert forecasts: here $\mu_1, \dots, \mu_n$ are n probability measure representing the forecasters' opinions (or probabilistic prediction if the expert is a statistical model). Another interesting setting is when data are released in an aggregated form due to privacy concerns (i.e., a data agency does not release microdata but just summaries in the form of unnormalized histograms).

The first challenge to face when approaching distributional data analysis (that is, analysis of data that are probability measures) is where to set the analysis: what is the space of which $\mu_1, \dots, \mu_n$ are points? Surely, if the μ_i's are probabilities over a space $\mathbb{X}$ (with σ-field $\mathcal{X}$), they must belong to $\mathbb{P}_{\mathbb{X}}$, the space of all probability measures over $\mathbb{X}$. We can endow (subsets of) $\mathbb{P}_{\mathbb{X}}$ with several metrics d, such as Hellinger distance:

$$2d_H^2(\mu, \nu) = \int_{\mathbb{X}} \left(\sqrt{\frac{d\mu}{d\lambda}} - \sqrt{\frac{d\nu}{d\lambda}} \right)^2 d\lambda$$

where $\frac{d\mu}{d\lambda}$ denotes the Radon-Nykodym derivative; integral probability metrics

$$d_{\mathfrak{F}}(\mu, \nu) = \sup_{f \in \mathfrak{F}} | \int f d\mu - \int f d\nu |$$

where $\mathfrak{F}$ denotes an appropriate class of functions from $\mathbb{X}$ to the reals; the Lévy-Prokhorov metric

$$d_{LP}(\mu, \nu) = \inf_{\varepsilon > 0} \{ \mu(A) \leq \nu(A^\varepsilon) + \varepsilon \text{ and } \nu(A) \leq \mu(A^\varepsilon) + \varepsilon \text{ for all } A \in \mathcal{X} \}$$

where A^ε is the ε-neighborhood of A. See, for instance, Gibbs and Su (2002) for a survey on probability metrics and their relations.

Then, upon considering $\mu_1, \dots, \mu_n$ as points in $(\mathbb{P}_{\mathbb{X}}, d)$, we can compute basic estimates such as the (Frechét) mean

$$\bar{\mu} = \inf_{\mu \in \mathbb{P}_{\mathbb{X}}} \sum_{i=1}^n d^2(\mu, \mu_i)$$

with the understanding that $d^2(\mu, \mu_i) = +\infty$ if the distance is not defined for the couple (μ, μ_i) (i.e., the case of $d \equiv d_H$, μ_i an absolutely continuous measure and μ a discrete one). It is clear that the choice of d highly impacts the output of the analysis: see Figure 10.0.1 for a comparison of the L_2 Frechét mean and the Wasserstein one (see Equation (10.1) below). Thus, the choice of d must reflect what one considers as the "mean" between to distributions. Consider for instance the case in Figure 10.0.1, and let us pretend that the blue and orange densities represent the heights of two populations. The L_2 mean

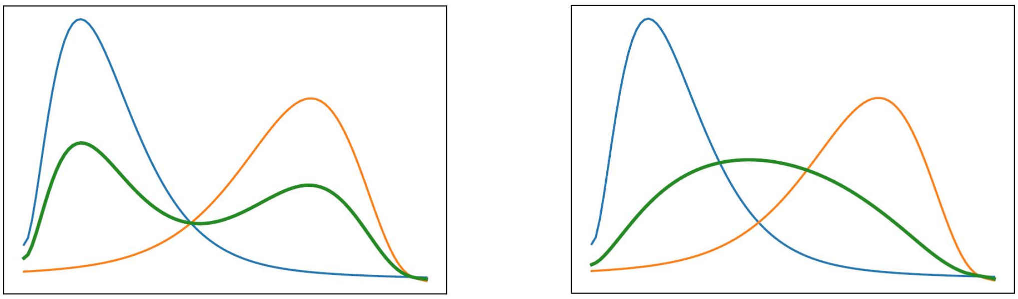


Figure 10.0.1: Frechét mean (green line) of two probability measures whose densities are depicted in blue and orange respectively, when d is the L_2 norm between the probability density functions (left plot) and when d is the Wasserstein metric (right).

corresponds to the distribution height of a third population obtained by mixing the blue and orange one. On the other hand, the Wasserstein distance is more similar to the law of a third population whose heights are "in between" the ones of the blue and orange populations. There is no overall "better" definition of barycenter and the choice of distance should be problem-specific.

Another important aspect to consider is the geometric structure of $(\mathbb{P}_{\mathbb{X}}, d)$. Indeed, a richer structure (such as a manifold structure or a linear one) naturally opens the door to more complex analyses. For example, principal component analysis (PCA) can be defined using the concept of *geodesic*. On the other hand, richer structures require more assumptions about the measures. As two opposite cases, consider $(\mathbb{P}_{\mathbb{X}}, d_{LP})$ and the Bayes space $B_2([a, b])$ of distributions with support $[a, b] \subset \mathbb{R}$ (Egozcue et al., 2006), with the inner product

$$\langle \mu, \nu \rangle := \frac{1}{2(b-a)} \int_a^b \int_a^b \log \frac{f_\mu(x)}{f_\mu(y)} \log \frac{f_\nu(x)}{f_\nu(y)} \mathrm{d}x\mathrm{d}y$$

where f_μ (f_ν) denotes the probability density function of μ (ν). This definition of inner product further makes $B_2([a, b])$ a Hilbert space.

It is well known that if $(\mathbb{X}, d_{\mathbb{X}})$ is a separable metric space, $(\mathbb{P}_{\mathbb{X}}, d_{LP})$ is also, and that d_{LP} metrizes the weak convergence topology. Hence, we can argue that d_{LP} is a very convenient distance to compare probability distributions: for instance, we can compare any two measures, even an absolutely continuous measure and an atomic one, which is often useful to establish certain limit theorems. However, numerical computation of d_{LP} is prohibitive, and thus it is not suited as a distance for distributional data analysis. On the other hand, the Hilbert structure of $B_2([a, b])$ makes it convenient to carry out computations and statistical analyses: the mean is trivially computed. Moreover, it is possible to define the covariance operator so that PCA can be computed via its singular values decomposition. However, the Bayes space encompasses only absolutely continuous probability distributions supported on the same interval, which is very restrictive.

In the next two chapters, we argue in favor of considering probability distributions as points of the *Wasserstein space*, endowed with the Wasserstein metric W_p (see, e.g., Ambrosio et al., 2008):

$$W_p^p(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{\mathbb{X} \times \mathbb{X}} d(x, y)^p \gamma(\mathrm{d}x\, \mathrm{d}y), \tag{10.1}$$

where $\Gamma(\mu, \nu)$ is the set of probability distributions over $\mathbb{X} \times \mathbb{X}$ with marginals μ and ν. Unlike the Lévy-Prokhorov metric, the p-th Wasserstein distance W_p is defined only on a subset of probability measures. However, this subset is quite large: it consists of the probability measures with finite "p-th moment", that is there exists $x_0 \in \mathbb{X}$ such that

$$\int_{\mathbb{X}} d(x, x_0)^p \mu(\mathrm{d}x).$$

In particular, it is possible to compare probability distributions with different supports and (10.1) does not require that the measures have densities. Moreover, the Wasserstein distance is a reasonably "weak" one as convergence in W_p is equivalent to weak convergence plus convergence of the p-th moment (Ambrosio et al., 2008).

The Wasserstein space is not Hilbert. Hence, classical statistical methods developed for multivariate data, such as linear regression and PCA, cannot be applied directly to distributions in the Wasserstein space. However, by exploiting its *weak Riemannian structure*, we can define meaningful statistical methods merging techniques developed for data living on Riemannian manifolds and for functional data. In particular, in Chapter 11 we consider the case of probability measures on the real line. By using a geometric structure closely related to the weak Riemannian structure of the 2-Wasserstein space, we propose a novel formulation of *projected* PCA and linear regression models. In Chapter 12 we consider probability measures on the circumference, inspired by a real dataset on eye nerve width's measurements. After establishing some results about optimal transport maps for measures on the circumference, we propose a framework for PCA.

11. Projected Statistical Methods for Distributional Data on the Real Line with the Wasserstein Metric

In this chapter, based on Pegoraro and Beraha (2022), we present a novel class of *projected* methods to perform statistical analysis on a data set of probability distributions on the real line, with the 2-Wasserstein metric. We focus in particular on Principal Component Analysis (PCA) and regression. To define these models, we exploit a representation of the Wasserstein space closely related to its weak Riemannian structure by mapping the data to a suitable linear space and using a metric projection operator to constrain the results in the Wasserstein space. By carefully choosing the tangent point, we are able to derive fast empirical methods, exploiting a constrained B-spline approximation. As a byproduct of our approach, we are also able to derive faster routines for previous work on PCA for distributions. By means of simulation studies, we compare our approaches to previously proposed methods, showing that our *projected* PCA has similar performance for a fraction of the computational cost and that the *projected* regression is extremely flexible even under misspecification. Several theoretical properties of the models are investigated, and asymptotic consistency is proven. Two real world applications to Covid-19 mortality in the US and wind speed forecasting are discussed.

11.1 Introduction

In many fields of machine learning and statistics, performing inference on a set of distributions is an ubiquitous but arduous task. The Wasserstein distance provides a powerful tool to compare distributions, as it requires very little assumptions on them and is at the same time reasonably easy to compute numerically. In fact, many other distances for distributions either require the existence of a probability density function or are impossible to evaluate, cf. Cuturi (2013), Peyré et al. (2019), Panaretos and Zemel (2020).

The Wasserstein distance recently gained popularity both in the statistics and machine learning community. See for instance Bassetti et al. (2006), Bernton et al. (2019), Catalano et al. (2021) for statistical properties of the Wasserstein distance, Cao et al. (2019), Cuturi et al. (2019) and Cuturi and Doucet (2014) for applications in the field of machine and deep learning, Bernton et al. (2019) and Srivastava et al. (2015a) for applications in Bayesian computation.

In this work, we focus on the situation in which the single observation itself can be seen as a distribution, as in the analysis of images (Cuturi and Doucet, 2014; Banerjee et al., 2015), census data (Cazelles et al., 2018), econometric surveys Potter et al. (2017) and process monitoring (Hron et al., 2014). In particular, we consider observations to be distributions on the real line. There exist several possible ways to represent distributions, such as histograms, probability density functions (pdfs) and cumulative density functions (cdfs), each characterized by different constraints. For instance, histograms sum to one, pdfs integrate to one, and the limits for cdfs are 0 and 1, moreover all of these functions are nonnegative. These constraints translate into complex geometrical structures that characterize the underlying spaces in which these objects live.

11.1.1 Previous work on distributional data analysis

One of the first works defining PCA for a data set of distributions is Kneip and Utikal (2001), where the authors apply tools from functional data analysis (FDA) directly to a collection of probability density functions. This approach, however, completely ignores the constrained nature of probability density functions, leading to poor interpretability of the results.

Based on theoretical results in Egozcue et al. (2006), who defines a Hilbert structure on a space of probability density functions on a compact interval (called a Bayes space), Delicado (2011) and Hron et al. (2014), propose a more reasonable approach to the problem of PCA for density functions. In particular, in Hron et al. (2014), the authors use the geometric properties of the Bayes space, coupled with a suitable transformation from the Bayes space to an L_2 space, to perform PCA on a set of pdfs using FDA tools, and then map back the results to the Bayes space.

Another, perhaps less widely used, approach focuses on borrowing tools from symbolic data analysis (SDA) in the context of histogram data (Nagabhushan and Pradeep Kumar, 2007; Rodríguez et al., 2000; Le-Rademacher and Billard, 2017). Moreover, in Verde et al. (2015) some of these attempts are extended to generic distributional data using Wasserstein metrics.

Finally, Bigot et al. (2017) and Cazelles et al. (2018) propose two PCA formulations based on the geometric structure of the Wasserstein space: a *geodesic* PCA and a *log* PCA. In a similar fashion, the recent works of Chen et al. (2021), Ghodrati and Panaretos (2021), and Zhang et al. (2020) propose regression and autoregressive models, respectively, for distributional data using the Wasserstein geometry.

We now highlight some key aspects of the aforementioned approaches. Hron et al. (2014) assumes that all the probability measures have the same support. This is hardly verified in practice, so that to apply their techniques one needs either to truncate the support of some of the probability density functions, or to extend others (for instance, by adding a small constant value and renormalizing), leading to numerical instability as discussed in Sections 11.7 and 11.8.

The SDA-based methods in Nagabhushan and Pradeep Kumar (2007); Rodríguez et al. (2000); Le-Rademacher and Billard (2017) and Verde et al. (2015) share the poor interpretability of SDA.

The methods in Bigot et al. (2017), Cazelles et al. (2018), Chen et al. (2021) and Zhang et al. (2020) are based on the weak Riemannian structure of the Wasserstein space, cf. Section 11.2.2. Such structure enables the authors to borrow ideas and terminologies from statistical frameworks defined on Riemannian manifolds (see Bhattacharya et al., 2012; Pennec, 2006, 2008; Huckemann et al., 2010; Patrangenaru and Ellingson, 2015; Fletcher, 2013; Banerjee et al., 2015). We can roughly distinguish those frameworks in two main approaches: the intrinsic/geodesic one and extrinsic/log one.

Briefly, intrinsic methods are defined using the metric structure of the Wasserstein space, working with geodesic curves and geodesic subsets, so that they faithfully respect the metric of the underlying space. However, in general, intrinsic methods present many practical difficulties in that the optimization problems they lead to are usually nontrivial, as we discuss in Section 11.5.3. Instances of intrinsic methods for distributional data are the *geodesic* PCA in Bigot et al. (2017) and, under some rather restrictive assumptions, the linear models in Chen et al. (2021) and the autoregressive models in Zhang et al. (2020), see Sections 11.3.3 and 11.3.4.

On the other hand, extrinsic methods resort to the linear structure of suitably defined tangent spaces, by mapping data from the Wasserstein space to the tangent (through the so-called *log* map) and then mapping back the results to the Wasserstein space (through the *exp* map). Of course, this approach is less respectful of the underlying geometry than the intrinsic one, but usually presents several numerical advantages. An example of such extrinsic methods defined in the Wasserstein space is the *log* PCA in Cazelles et al. (2018).

The main issue with this *log* PCA is that the image of the *log* map inside the tangent of the Wasserstein space is not a linear space, but rather a convex cone embedded in a linear space (see Section 11.2.2). Hence, while exploiting the linear structure of the tangent, it is possible that the projection of some points onto the principal components end up outside the cone. For these points, the *exp* map from the tangent to the Wasserstein space used in Cazelles et al. (2018) is not a metric projection, which in general is not available, so that the results in this setting are hardly interpretable.

11.1.2 Our contribution and outline

The contribution of this work is three folded. First, we propose alternative PCA and regression models for distributional data in the Wasserstein space. We term these models *projected*, in opposition to the *log* PCA in Cazelles et al. (2018). Second, by exploiting a geometric characterization of Wasserstein space closely related to its weak Riemannian structure, we build a novel approximation of the Wasserstein space using monotone B-spline. This allows us to represent the space of probability measures as a convex polytope in $\mathbb{R}^J$. Lastly, we obtain faster optimization routines for the *geodesic* PCAs defined in Bigot et al. (2017), exploiting the aforementioned B-spline representation.

Our *projected* framework lies in between the *log* one and the *geodesic* one, since we use an analogous to the *log* map to transform our data, as for extrinsic methods, but do not resort to the *exp* map to return to the Wasserstein space, using instead the metric projection operator. Thanks to this, our *projected* methods are more respectful of the underlying geometry than the *log* ones, while at the same time retaining the same reduced computational complexity. Thus, the *projected* methods expand the range of situations where *extrinsic* methods are an effective and efficient alternative to intrinsic tools: in our examples, the performance loss in general is marginal (see Section 11.7).

By centering the analysis in appropriate points of the Wasserstein space, one can identify the space of probability measures (with finite second moment) with the space of square-integrable monotonically non-decreasing functions on a compact set. We use a suitable quadratic B-spline expansion to get a very handy representation of such functions. Through such B-spline expansion, it is possible to approximate the metric projection onto the Wasserstein space as a constrained quadratic optimization problem over a convex polytope, that is a well-established problem, cf. Potra and Wright (2000). This allows us to exploit the underlying linear structure of an L_2 space, so that all the machinery developed for functional data analysis can be directly applied to this setting. We address the issue of interpretability of the results, tackling a number of diverse applications and developing different ways to measure the loss of information caused by the *extrinsic* nature of our methods.

We observe that the idea of representing nondecreasing functions through B-splines for statistical purposes has been proposed also by Das and Ghosal (2017), in the context of Bayesian quantile regression, where the authors use B-splines with (random) monotonic coefficients as a generative model for random quantile functions. However, their focus is on defining a generative model, and not on developing a statistical setting exploiting the geometry given by the constrained representation. Along this direction, they do not restrict their attention to quadratic splines and consider cubic ones.

As already mentioned, a further contribution of this work is the derivation of alternative numerical optimization schemes for the *geodesic* PCA in Bigot et al. (2017) and Cazelles et al. (2018), based on the proposed quadratic B-spline expansion.

The remaining of the chapter is organized as follows. Section 11.2 covers the basic concepts of Wasserstein distance and the weak Riemannian structure of the Wasserstein space, along with a brief discussion on a suitable way to exploit such structure for our purposes. Section 11.3 defines the *projected* PCA and *projected* regression in a general setting. In Section 11.4 we discuss the choice of the base point in which we center our analysis and how to efficiently approximate the metric projection through B-splines; in Section 11.5 we present the numerical algorithms needed to compute our *projected* methods and an alternative optimization routine for the *geodesic* PCA in Cazelles et al. (2018). Section 11.6 discusses the asymptotic properties of the spline approximation and of the *projected* models, establishing consistency of the estimators under some assumptions. Numerical illustrations on real and simulated data sets are shown in Sections 11.7 and 11.8. In particular, we apply our projected methods to two real world problems: we perform PCA on the US data on Covid-19 mortality by age and sex and perform a distribution regression to forecast the wind speed near a wind farm. Finally, the article concludes in Section 11.9. The Appendix collects all the proofs of the theoretical results, additional details on the simplicial PCA and regression, and further simulations. Code for reproducing the numerical results is available at https://github.com/mberaha/ProjectedWasserstein.

11.2 Preliminaries

In the following, we will consider probability measures on the real line $\mathbb{R}$ endowed with the usual Borel σ-field, we will skip references to the σ-field whenever it is obvious.

Given a measure μ on $\mathbb{R}$ define its cumulative distribution function $F_\mu(x) = \mu((-\infty, x])$ for $x \in \mathbb{R}$ and the associated quantile function $F_\mu^-(t) = \inf\{x \in \mathbb{R} : \ t \leq F_\mu(x)\}$. When F_μ is continuous and strictly monotonically increasing, $F_\mu^- = (F_\mu)^{-1}$.

11.2.1 Wasserstein metric and Wasserstein spaces

We start by recalling the definition of the 2-Wasserstein distance between two probability measures μ, ν on $\mathbb{R}$:

$$W_2^2(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{R}\times\mathbb{R}} |x-y|^2 d\gamma(x,y), \tag{11.1}$$

where $\Gamma(\mu, \nu)$ is the collection of all probability measures on $\mathbb{R} \times \mathbb{R}$ with marginals μ and ν. Closely related to the definition of Wasserstein distance lies the one of Optimal Transport (OT). In particular, (11.1) identifies the Wasserstein distance with the minimal total transportation cost between μ and ν in the Kantorovich problem with quadratic cost (Ambrosio et al., 2008).

For our purposes, it is convenient to consider another formulation of the OT problem, originally introduced by Monge (1781). Given two measures μ, ν as before, the optimal transport map from μ to ν is the solution of the problem

$$\inf_{T : T\#\mu=\nu} \int_\Omega |x - T(x)|^2 d\mu(x), \tag{11.2}$$

where # denotes the pushfoward operator, that is for any measurable set B and measurable function

$$f : \mathbb{R} \to \mathbb{R}, \qquad (f\#\mu)(B) = \mu(f^{-1}(B)). \tag{11.3}$$

Note that any solution of (11.2) induces one and only one solution of (11.1); moreover if the OT problem has a unique solution, then also the Wasserstein distance problem has only one solution. However not all Wasserstein distance problems can be solved through Monge's formulation (Ambrosio et al., 2008).

The unidimensional setting is a remarkable exception in that there exist explicit formulas for both problems. In particular, the Wasserstein distance can be computed as

$$W_2^2(\mu,\nu) = \int_0^1 |F_\mu^-(s) - F_\nu^-(s)|^2 ds, \tag{11.4}$$

and, if the measure μ has no atoms, then there exists a unique solution to Monge's problem given by $T_\mu^\nu = F_\nu^- \circ F_\mu$. For a proof of these results, see Chapter 6 of Ambrosio et al. (2008).

It is clear that, in general, the Wasserstein distance between two probability measures can be unbounded (for instance when in (11.4) F_μ^- is not square-integrable on $[0,1]$). Nonetheless, when restricting the focus on the set of probability measures with finite second moment, then it holds that W_2 defines a metric (see, for instance, Chapter 7 of Villani, 2008). Formally, let the Wasserstein space:

$$\mathcal{W}_2(\mathbb{R}) = \Big\{\mu \in \mathcal{P}(\mathbb{R}) \ : \int_{\mathbb{R}} x^2 d\mu < +\infty\Big\},$$

then $(\mathcal{W}_2(\mathbb{R}), W_2)$ is a separable complete metric space.

11.2.2 Weak Riemannian structure of the Wasserstein Space

Thanks to the uniqueness of the transport maps, by fixing an absolutely continuous (a.c.) probability measure $\mu \in \mathcal{W}_2(\mathbb{R})$, we can associate to any $\nu \in \mathcal{W}_2(\mathbb{R})$ the optimal transport map T_μ^ν. Since $\int_{\mathbb{R}} |T_\mu^\nu(x)|^2 d\mu = \int_{\mathbb{R}} x^2 d\nu$ we can define the following map $\varphi_\mu : \mathcal{W}_2(\mathbb{R}) \to L_2^\mu(\mathbb{R})$ with the rule: $\varphi_\mu(\nu) = T_\mu^\nu$.

We note several immediate but interesting properties of the map φ_μ. First, it is an isometry (and so a homeomorphism onto its image) since

$$\int_{\mathbb{R}} |T_\mu^\nu(x) - T_\mu^\eta(x)|^2 d\mu = \int_{[0,1]} |F_\nu^-(s) - F_\eta^-(s)|^2 ds = W_2^2(\nu,\eta).$$

Second, the image of φ_μ is a closed convex cone in $L_2^\mu(\mathbb{R})$: a set closed under addition and positive scalar multiplication. In fact, for any $\lambda \geq 0$, λT_μ^ν is still a transport map from μ to another measure whose quantile is λF_μ^-; and similarly $T_\mu^\nu + T_\mu^\eta = (F_\nu^- + F_\eta^-) \circ F_\mu$. Being $\mathcal{W}_2(\mathbb{R})$ complete, $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ is closed in $L_2^\mu(\mathbb{R})$. Third, $\varphi_\mu(\mu) = id_{\mathbb{R}}$ (where id_C denotes the identity map of the set C). Finally, as shown in Panaretos and Zemel (2020), φ_μ is not surjective and $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ is the set of μ-a.e. non decreasing functions in $L_2^\mu(\mathbb{R})$.

The inverse of the map of φ_μ is the measure pushforward (see Equation 11.3) and it is defined on the whole $L_2^\mu(\mathbb{R})$: given $f \in L_2^\mu(\mathbb{R})$, then $\nu = f\#\mu$ is a measure in $\mathcal{W}_2(\mathbb{R})$. In fact:

$$\int |x|^2 d\nu = \int |f(x)|^2 d\mu = \|f\|_\mu^2.$$

A natural way to define a tangent structure for $\mathcal{W}_2(\mathbb{R})$ is therefore to take advantage of the cone structure given by φ_μ. In fact, for closed convex cones, there are already notions of tangent cones. Similarly to Rockafellar and Wets (1998), Theorem 6.9, we can define:

$$\mathrm{Tan}_\mu(\mathcal{W}_2(\mathbb{R})) := \mathrm{Tan}_{id_{\mathbb{R}}}(L_2^\mu(\mathbb{R})) = \overline{\{f \in L_2^\mu(\mathbb{R}) | \exists h > 0 : id + hf \in \varphi_\mu(\mathcal{W}_2(\mathbb{R}))\}}^{L_2^\mu(\mathbb{R})}. \tag{11.5}$$

We remark that Theorem 6.9 in Rockafellar and Wets (1998) is stated in $\mathbb{R}^n$, but it holds also more generally, for instance in an Hilbert space (see Aubin and Frankowska (2009), Chapter 4).

A geometric interpretation of (11.5) is the following. The tangent space consists of all the vectors f that move the base point inside the cone $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$, when considered up to a scale factor h. Hence, f plays the role of direction of a tangent vector going out from the tangent point. Furthermore, since for every $f \in \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ then $f + id \in \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ we have that $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ is included in the tangent space. As shown later in this Section, the inclusion is strict and the tangent space is much larger than $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$.

Note that we can recover the definition of tangent space given by Ambrosio et al. (2008) and Panaretos and Zemel (2020) by a simple 'change of variable': calling $g = id + hf$ then substituting $(g - id)/h$ in (11.5) gives the following definition of tangent

$$\mathrm{Tan}_\mu(\mathcal{W}_2(\mathbb{R})) = \overline{\{\lambda(f - id) | f \in \varphi_\mu(\mathcal{W}_2(\mathbb{R})); \lambda > 0\}}^{L_2^\mu(\mathbb{R})},$$

which is the one given in Ambrosio et al. (2008) and Panaretos and Zemel (2020). As shown in Panaretos and Zemel (2020) the tangent cone $\mathrm{Tan}_\mu(\mathcal{W}_2(\mathbb{R}))$ is indeed a linear space. For this reason we refer to it as tangent space, instead of cone.

In analogy to Riemannian geometry, following Ambrosio et al. (2008) and Panaretos and Zemel (2020), we define the $\log_\mu$ and $\exp_\mu$ maps. Having fixed μ absolutely continuous:

$$\begin{aligned} \log_\mu : \mathcal{W}_2(\mathbb{R}) &\to \mathrm{Tan}_\mu(\mathcal{W}_2(\mathbb{R})) & \qquad \exp_\mu : \mathrm{Tan}_\mu(\mathcal{W}_2(\mathbb{R})) &\to \mathcal{W}_2(\mathbb{R}) \\ \nu &\mapsto T_\mu^\nu - id & f &\mapsto (id + f)\#\mu \end{aligned} \tag{11.6}$$

We briefly highlight some properties of these maps, which immediately follow from the discussion above.

Remark 11.1. *The map $\log_\mu$ is defined on the whole space $\mathcal{W}_2(\mathbb{R})$. Moreover, it is clearly an isometry: $W_2(\eta, \nu) = \| \log_\mu(\eta) - \log_\mu(\nu)\|_{L_2^\mu(\mathbb{R})}$ (Panaretos and Zemel, 2020). This shows that there is no local-approximation issue when working in the tangent space, in contrast with the usual Riemannian manifold setting. There, the tangent space usually provides good approximation only in a neighborhood of the tangent point.*

Remark 11.2. *The map $\log_\mu$ is not surjective on Tan$_\mu$, indeed its image Im$(\log_\mu)$ is a closed convex subset of $L_2^\mu(\mathbb{R})$ given by all the maps f such that $f + id \in \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$, that is, $f + id$ is μ-a.e. increasing. The restriction of $\exp_\mu$ on Im$(\log_\mu)$, henceforth denoted by $\exp_{\mu|\log_\mu(\mathcal{W}_2(\mathbb{R}))}$, is an isometric homeomorphism and its inverse is $\log_\mu$. In particular, we observe that $\log_\mu \circ \exp_\mu$ is not a metric projection in L_2^μ. That is, in general $\log_\mu \circ \exp_\mu(f) \neq \arg\min_{g \in Im(\log_\mu)} ||f - g||_{L_2^\mu}$.*

11.2.3 Intrinsic and extrinsic methods in the Wasserstein space

As mentioned in Section 11.1.1, borrowing ideas from Riemannian geometry leads to discerning statistical methods on the Wasserstein space in the classes of *intrinsic* and *extrinsic* methods.

The Weak Riemannian structure presented in Section 11.2.2 provides a suitable environment for developing intrinsic methods. In fact, the geodesic structure of $\mathcal{W}_2(\mathbb{R})$ can be recovered through the linear structure of any $L_2^\mu(\mathbb{R})$ space through the isometry φ_μ. Pointwise interpolation of the transport maps coincide with the geodesic between measures. In other words, given μ a.c., the geodesic between ν and η is given by:

$$\gamma(t) = ((1 - t) \cdot T_\mu^\nu + t \cdot T_\mu^\eta)\#\mu. \tag{11.7}$$

Thus, such geodesic structure can be recovered in many different (but equivalent) ways, depending on μ.

On the other hand, Remark 11.1 motivates the development of extrinsic tools, since working in the image of $\log_\mu$ inside the tangent space Tan$_\mu$ is exactly like working in $\mathcal{W}_2(\mathbb{R})$. This is not common in Riemannian manifold framework, since usually the tangent space provides a good approximation only near the tangent point. As a consequence, if in the general Riemannian manifold framework the choice of the tangent point μ is crucial (since results for extrinsic methods might be significantly altered for different choices of μ) when working with $\mathcal{W}_2(\mathbb{R})$ this is not the case.

To further motivate this key point, consider μ and ν a.c. measures; the maps $\log_\nu \circ(\exp_{\mu|\log_\mu(\mathcal{W}_2(\mathbb{R}))})$ and $\varphi_\nu \circ \varphi_\mu^{-1}$ are isometric homeomorphisms (as composition of isometries and homeomorphisms). In other words, they preserve distances and send border elements of $\log_\mu(\mathcal{W}_2(\mathbb{R}))$ or $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ into border elements of $\log_\nu(\mathcal{W}_2(\mathbb{R}))$ and $\varphi_\nu(\mathcal{W}_2(\mathbb{R}))$, respectively, and the same with internal points (and so in particular, they preserve distances from any point to the border). In Chen et al. (2021), Bigot et al. (2017) and Zhang et al. (2020) μ is chosen as the barycentric measure $\bar{x}$ of the observations $x_i \in \mathcal{W}_2(\mathbb{R})$. The discussion above implies that considering the tangent space at the Wasserstein barycenter $\bar{x}$ and working on $\log_{\bar{x}}(x_i) = \log_{\bar{x}}(x_i) - \log_{\bar{x}}(\bar{x})$ is exactly the same as considering the tangent space at any μ a.c. and working on $\log_\mu(x_i) - \log_\mu(\bar{x})$ for our statistical purposes. So the choice of the tangent space from the theoretical point of view is completely arbitrary. Moreover, centering the analysis in the barycenter presents a drawback when studying asymptotic properties of the models under consideration, since $\bar{x}$ changes as the sample size grows.

In Section 11.4.1 we propose to fix μ as the uniform measure on $[0, 1]$. This choice not only allows us to derive empirical methods that are extremely simple to implement, cf. Section 11.5, but also allows us to study asymptotic properties of the models in Section 11.6.2 without resorting to parallel transport, as done for instance in Chen et al. (2021).

11.2.4 Tangent vs. L_2^μ

Lastly, we briefly discuss the major differences between using a tangent space representation of $\mathcal{W}_2(\mathbb{R})$ and using the representation given by some φ_μ.

We recall that, for a fixed μ a.c., the two representations are indeed quite similar $\varphi_\mu(\nu) = T_\mu^\nu$, $\log_\mu(\nu) = T_\mu^\nu - id$; a priori one may prefer the tangent representation, because it already expresses data as vectors coming out of a point. Therefore, for instance, it might result practically more convenient to center the analysis in the barycenter and work on vectors, taking away any 'data centering' issues. At the same time, also notational coherence with already existing methods might benefit from this choice.

However, especially when dealing with extrinsic techniques, we found slightly more practical to use the φ_μ representation in that it is more straightforward to represent $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ compared to $\log_\mu(\mathcal{W}_2(\mathbb{R}))$: the first one can in fact be represented directly as the cone of the μ-a.e non-decreasing functions.

11.3 Projected Models in the Wasserstein Space

In this section, exploiting the embeddings given by φ_μ, we define a class of *projected* statistical methods to perform extrinsic analysis for data in the Wasserstein space.

To give a general framework, we do not restrict our attention to a particular φ_μ yet, even though in Section 11.4 we argue that a natural choice which allows for an easier implementation of the empirical methods is letting μ be the uniform distribution on $[0, 1]$. Hence, for the sake of notation, we consider a generic case of data lying in a closed convex cone X inside a separable Hilbert space H. In our setting, H would be $L_2^\mu(\mathbb{R})$ and $X = \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$, for some $\mu \in \mathcal{W}_2(\mathbb{R})$ absolutely continuous.

11.3.1 Principal component analysis

We start by defining one of the main contributions of our work: the *projected* PCA. We recall that for an H-valued random variable $\mathcal{X}$, PCA is a well established technique and amounts to finding the eigenfunctions of the Karhunen-Loéve expansion of the covariance operator of $\mathcal{X}$, see Ramsay (2004). Observe that any X-valued random variable can be considered as an H-valued one (by the inclusion map), so that a notion of PCA is already available.

When defining principal components, a key notion is the one of dimension of the principal component (PC). In this work, principal components will be closed convex subsets of H, and we will always define the dimension of a subset of H as the dimension of the smallest affine subset of H containing it. For a generic closed convex set $C \subset H$, let Π_C denote the metric projection onto C: $\Pi_C(x) := \arg\min_{c \in C} ||x - c||$ and, for a set of vectors U, denote with $Sp(U)$ its linear span.

In what follows, we denote by x_0 the 'center' of the PCA. For us, $x_0 = \mathbb{E}[\mathcal{X}]$, or its empirical counterpart. To have a well defined PCA, we always assume that x_0 belongs to the relative interior of the convex hull of the support of $\mathcal{X}$, see Appendix 11.A for the definition of relative interior and further details. This is a rather technical hypothesis but it is not a restrictive one. For instance, it is always verified for empirical measures and when $X \subseteq \mathbb{R}^d$ and hence for our empirical methods, cf. Section 11.5.1.

Definition 1. *(Projected PCA). Given $\mathcal{X}$ a random variable with values in $X \subset H$, let $U_k = \{w_1, ..., w_k\}$ be its first k H-principal components centered in $x_0 = \mathbb{E}[\mathcal{X}]$. A $(k, x_0)-$projected principal component of $\mathcal{X}$ is the biggest closed convex subset $U_X^{x_0,k}$ of X such that: (i) $x_0 \in U_X^{x_0,k}$, (ii) $dim(U_X^{x_0,k}) = k$, and (iii) $U_X^{x_0,k} \subseteq \Pi_X(Sp(U_k))$.*

In other words, the projected principal component is obtained by approximating the span of the principal components found in H, with convex subsets in X. Note that the principal components in H might 'capture' some variability which is not present when measuring distances inside X. In fact, the projection of a point belonging to X onto a direction w_j might end up being outside X, see Section 11.3.3. However, as we will show in Section 11.7, in our examples the projected PCA behaves well and this issue does not seem to affect significantly the performance.

Remark 11.3. *Convex sets are essential in our analysis since, thanks to* (11.7)*, convex sets in X are precisely the subsets of $\mathcal{W}_2(\mathbb{R})$ which are geodesically complete: the geodesic connecting any pair of points in the subset is contained in the subset. Geodesic subsets are a natural generalization of linear spaces.*

Remark 11.4. *The metric projection of a linear subspace onto a convex subset can end up being a nonconvex set. In addition to that, while losing convexity, the dimension of the metric projection of a convex subset can be bigger than the dimension of the original subset. A simple example where both cases happen is the projection of $y = -x$ onto $x, y \geq 0$ in $\mathbb{R}^2$.*

We observe that inside a projected principal component, we have a preferential orthonormal basis given by the principal components in H; for this reason, we call $U_k = \{w_1, ..., w_k\}$ *principal directions.*

Although it might seem impractical to find the projected component, the following lemma provides a more convenient alternative characterization.

Lemma 11.1. *Let x_0 and $U_X^{x_0,k}$ be as in Definition 1, then $U_X^{x_0,k} = (x_0 + Sp(U_k)) \cap X$.*

Natural alternatives to Definition 1 would be, for instance, to let the projected principal directions (component) be the metric projection of $w_1, \dots, w_k$ (the linear span of $\{w_1, \dots, w_k\}$) onto X, respectively. In the former case, the projection would not guarantee the orthogonality of the projected directions, which is instead essential to properly explore the variability. Moreover, since the 'tip' of the projected unit vectors would likely lie on the border of X, the projection of a new observations on a direction would still lie outside of X as soon as the score associated to that direction is larger than 1. The latter case, instead, presents the drawbacks pointed out in Remark 11.4.

We argue that, despite its simplicity, Definition 1 is indeed very well suited for statistical analysis in the Wasserstein Space. For instance, we are guaranteed that, as the dimension grows up, the k projected components provide a monotonically better fit to the data. This is easily verified because Π_X is a strictly non-expansive operator, being X closed and convex (see Deutsch (2012)), which implies the following proposition.

Proposition 11.1. *With the same notation as Definition 1, for any $x \in X$ we have:*

$$\|\Pi_{U_X^{x_0,k}}(x) - x\| \geq \|\Pi_{U_X^{x_0,k+1}}(x) - x\| \to 0 \text{ with } k \to +\infty.$$

Once a principal component is found, a classical task that one may want to perform is to project a new 'observation' $x^* \in X$ onto $U_X^{x_0,k}$, for instance, for dimensionality reduction purposes. In general, the metric projection on generic convex subsets might be arduous to find, we will deal with this issue in Section 11.4. Nevertheless, we can use the following proposition to reduce in advance the dimension of the parameters involved in the problem; turning it into a projection problem inside the principal projected component, which allows for faster computations (see Equation 11.13).

Proposition 11.2. *Let $x^* \in X$ and let Π_k be the orthogonal projection on Span(U_k). The projection of x^* onto $U_X^{x_0,k}$ is given by*

$$\underset{v' \in U_X^{x_0,k}}{\arg\min} \|x^* - v'\| = \Pi_{Sp(U_k) \cap (X - x_0)}(\Pi_k(x^* - x_0)) + x_0. \tag{11.8}$$

Lastly, we observe that, since projected principal components are not linear subspaces, the scores of some points on a principal direction can vary as we increase the dimension of the principal component.

11.3.2 Regression

Broadly speaking, a regression model between two variables with values in two different spaces is given by an operator between such spaces, which for every input value of the independent variable returns a predicted value for the dependent variable. In the following, let us denote with $\mathcal{Z}$ the independent variable and with $\mathcal{Y}$ the dependent one. A regression model is usually understood as an operator Γ specifying the conditional value of $\mathcal{Y}$ given $\mathcal{Z}$, that is, $\mathbb{E}[\mathcal{Y}|\mathcal{Z}] = \Gamma(\mathcal{Z})$.

If the spaces where $\mathcal{Z}$ and $\mathcal{Y}$ take values possess a linear structure, this linearity is usually exploited by means of a (kernel) linear operator, with possibly an 'intercept' term. To define our *projected* regression model, we want to exploit the cone structure of X in a similar fashion. In fact, such linear kernel operators combine good optimization properties and interpretability since their kernels can provide insights into the analysis, much like coefficients in multivariate linear regression.

We treat separately the cases where the X-valued variable is the independent or the dependent one. The case when both variables are X-valued follows naturally. To keep the notation light, in what follows, we will not distinguish between 'proper' linear operators and linear operators with an added intercept term, which could as well be employed in all the incoming definitions to gain flexibility.

Consider the case in which we have an independent X-valued random variable and denote with V the space where the dependent variable takes values. Despite the fact that X is not a linear space, with an abuse of notation, we call 'linear' an operator which respect sum and positive scalar multiplication for elements in X. Such operators are indeed obtained by restricting on X linear operators defined on H. Following this idea, in order to define linear regression for an X-valued independent random variable, we consider such variable as H-valued, obtain the regression operator and then take the restriction of the operator on X. In this way, when $H = L_2^{\mu}(\mathbb{R})$ and $X = \varphi_{\mu}(\mathcal{W}_2(\mathbb{R}))$, it is possible to exploit the classical FDA framework to perform all kinds of distribution on scalar/vector/etc... regression. For brevity, we report only the definition with $V = \mathbb{R}$.

Definition 2. *Let $\mathcal{Z}$ an X-valued random variable, and $\mathcal{Y}$ a real valued one. Let $\Gamma_{\beta} : H \to \mathbb{R}$ be a functional linear regression model for such variables, with $\mathcal{Z}$ considered as H-valued and $\Gamma_{\beta}(v) = \langle \beta, v \rangle$. A projected linear regression model for $(\mathcal{Z}, \mathcal{Y})$ is given by $(\Gamma_{\beta})_{|X}$.*

Now we turn to the cases which feature an X valued dependent variable and a Z valued independent one, for Z a generic Hilbert space. Through the inclusion $X \hookrightarrow H$, we can consider a regression problem with X-valued dependent variable as a problem with H-valued dependent variable. Comparing this situation with the previous one, it is clear that we now face a 'dual' problem. Indeed, while before we needed to restrict the domain from H to X, we now need to force the codomain of Γ to lie inside X. We would like to retain the same properties that make linear kernel operators appealing as regression operators between Hilbert spaces. A possibility could be considering a linear kernel operator Γ with values in H and restricting it to $\Gamma^{-1}(X)$. However, this would imply that for any $z \notin \Gamma^{-1}(X)$ no prediction would be available.

We argue that a more reasonable approach consists in finding an operator $\Gamma_{\mathrm{P}} : Z \to X$ as close as possible (in some sense that will be clear later) to the linear kernel operator Γ aforementioned. Hence, we relax the linearity assumption in favor of Lipschitzianity and take as regression operator $\Pi_X \circ \Gamma$, whose image always lies in X. Note that Γ_{P} inherits the interpretability of the kernel of Γ.

To motivate such a choice, we give the following notion of a projected operator.

Definition 3. *Let Z be a normed space and consider $\mathcal{Z}$ a Z-valued random variable. Let $\Gamma : Z \to H$ a generic Lipschitz operator between Z and H. A $(\mathcal{Z}, X)$-projection of Γ is an operator $\Gamma_{\mathrm{P}} : Z \to X$ such that:*

$$\Gamma_{\mathrm{P}} = \underset{T:Z \to X}{\arg\min}\, \mathbb{E}_{\mathcal{Z}}[\|\Gamma(v) - T(v)\|^2].$$

In other words, Γ_{P} provides the best pointwise approximation of the H-valued operator Γ, averaged w.r.t. the measure induced by $\mathcal{Z}$. Hence, given $\mathcal{Z}$, a Z-valued random variable, $\mathcal{Y}$, an X-valued random variable, and a linear regression model $\Gamma : Z \to H$ for $(\mathcal{Z}, \mathcal{Y})$, the projected regression model induced by Γ is Γ_{P}.

Proposition 11.3. *With the same notation as above, if $\mathbb{E}\left[\|\mathcal{Z}\|^2\right] < \infty$, then $\Gamma_{\mathrm{P}} = \Pi_X \circ \Gamma$.*

Proof. For any $T : Z \to X$, it holds: $\|\Gamma(z) - \Pi_X(\Gamma(z))\| \leq \|\Gamma(v) - T(v)\|$. Moreover, Γ and $\Pi_X \circ \Gamma$ are Lipschitz, and being Π_X non-expansive, they share the same constant $L > 0$:

$$\|\Gamma(v) - \Pi_X \circ \Gamma(v)\|^2 \leq 2L\|v\|^2$$

and thus $\mathbb{E}_{\mathcal{Z}}[\|\Gamma(z) - \Pi_X \circ \Gamma(z)\|^2]$ is bounded iff $\mathcal{Z}$ has finite second moment. □

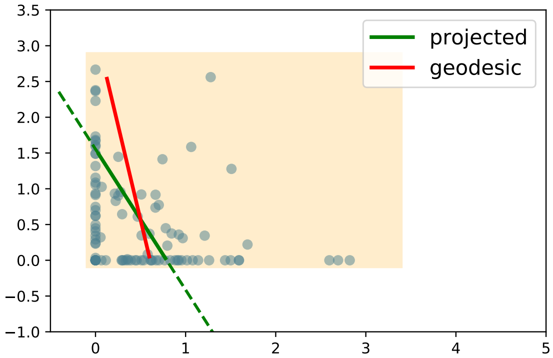


Figure 11.3.1: Comparison of projected and geodesic PCA when $H = \mathbb{R}^2$ and X is the shaded rectangle. The projected principal direction is rather different from the geodesic one because most of the observations (blue dots) are concentrated around the borders

The only case left out from the treatment above is when both the independent and the dependent variables are X-valued. This case, however, follows naturally by combining the two approaches and we report the definition below.

Definition 4. *Let $\mathcal{Z}$ and $\mathcal{Y}$ two X-valued random variables. Let $\Gamma : H \to H$ be a functional linear regression model for the variables considered as H-valued. A projected linear regression model for $(\mathcal{Z}, \mathcal{Y})$ is given by $(\Pi_X \circ \Gamma)_{|X}$.*

Remark 11.5. *When considering a regression with X-valued independent variable, one may want to relax the restriction on X in Definition 2 for various reasons; for instance, one may have measurement errors, or, by design, the test set may consider points also outside X. In such cases, it is worth considering the problem of how many continuous linear extensions of $\Gamma_{|X}$ are possible on the whole H. A sufficient condition for the uniqueness of such extension is the following: there exist a sequence of linear subspaces of H, say $\{H_J\}_{J\geq 1}$, such that $\bigcup_J H_J$ is dense in H and $X_J := H_J \cap X$ contains a basis of H_J for every J.*

Remark 11.6. *When $H = L_2^\mu(\mathbb{R})$ and $X = \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ the condition in Remark 11.5 is verified, for instance, by Remark 11.8 in Section 11.4.3. Moreover, observe that the uniqueness of the extension can also be proven thanks to Jordan's representation of functions $f : \mathbb{R} \to \mathbb{R}$ with bounded variation (BV). In fact, any f with BV can be written as the difference of monotone functions and thus $\Gamma(f)$ is fixed. Then by the density of BV functions in H, we define Γ on the remaining elements of H.*

11.3.3 Comparison with intrinsic methods

We now compare the projected methods defined earlier in this Section and the intrinsic counterparts. In particular, we focus on the *geodesic* PCA defined in Bigot et al. (2017) and Cazelles et al. (2018) and on the distribution on distribution regression model in Chen et al. (2021).

Bigot et al. (2017) and Cazelles et al. (2018) define two different PCA, namely a global and a nested one; in particular the nested approach presents analogies with other PCAs developed for manifold valued random variables (Jung et al., 2012; Huckemann and Eltzner, 2018; Pennec, 2018); we report the two definitions below.

Definition 5. *(Global geodesic PCA) Let $\mathcal{X}$ a random variable with values in X with $\mathbb{E}[\mathcal{X}] = x_0$. A (k, x_0)-global geodesic PC is a set C^* minimizing $\mathbb{E}\left[d(\mathcal{X}, C)^2\right]$ over the closed convex sets $C \subset X$ such that $x_0 \in C$ and $dim(C) \leq k$.*

Definition 6. *(Nested geodesic PCA) Let $\mathcal{X}$ a random variable with values in X with $\mathbb{E}[\mathcal{X}] = x_0$. For $k = 1$, a (k, x_0)-nested geodesic PC is a set C_k^* such that C_k^* is a minimizer of $\mathbb{E}\left[d(\mathcal{X}, C)^2\right]$ over the closed convex sets $C \subset X$ such that $x_0 \in C$ and dim$(C) \leq k$; for $k \geq 1$, a (k, x_0)-nested geodesic PC is a set C_k^* such that C_k^* is a minimizer of $\mathbb{E}\left[d(\mathcal{X}, C)^2\right]$ over the closed convex sets $C \subset X$ such that: $x_0 \in C$, dim$(C) \leq k$, and $C \supset C_{k-1}^*$, where C_{k-1}^* is a $(k-1, x_0)$-nested geodesic PC.*

The first key difference between the global and the nested geodesic PCA is that the latter provides a notion of preferential directions in the principal component, while the first one does not. In fact, the first nested principal component corresponds to the first principal direction, and it is possible to find the remaining principal directions by imposing orthogonality constraints as we obtain nested PCs of higher dimensions. Thus, the nested geodesic PCA is more suitable to explore and visualize the variability in a data set, see also Section 11.7. On the other hand, exactly because of the lack of such constraints, the global PCA is, in general, more flexible and provides superior performance in terms of *reconstruction error*, cf. Section 11.7.

Comparing these definitions with the one of our projected PCA, the key difference is that geodesic PCAs do not exploit the Hilbert structure of H. Thus, as we discuss in Section 11.5.3, the numerical routines needed to find such principal components rely on nonlinear constrained optimization, which can be extremely demanding and nontrivial to implement. This is in sharp contrast with our projected PCA in Definition 1, that, thanks to Lemma 11.1 can be straightforwardly computed. However, as a result, the projected PCA is in general less respectful of the underlying metric structure. By investigating this issue in simpler settings, for instance, when $H = \mathbb{R}^d$ and X is a convex polytope in $\mathbb{R}^d$, we noticed that the differences between the projected principal directions and the nested geodesic ones become appreciable only if the random variable $\mathcal{X}$ gives significant probability to values near the borders of X. See for instance Figure 11.3.1.

Note that the interpretability of the projected PCA is determined by the level of discrepancy between the projected and nested principal directions, as in Figure 11.3.1, which depends on how much variability it is correctly captured by the component, that is, how much of the variability captured by the projected component lies in X. This intuition is formalized in Section 11.7.2 where two measures of 'reliability' of the projected PCA are proposed.

Turning to the regression context, Chen et al. (2021) define a distribution on distribution linear regression model in the Wasserstein space. Their approach considers two different tangent spaces of $\mathcal{W}_2(\mathbb{R})$ (the first one centered in the barycenter of the independent variable and the second one centered in the barycenter of the dependent variable) and map the observations to the corresponding tangent spaces. They then use FDA tools to estimate a functional linear model $\widehat{\Gamma}$ between those two spaces. When the image of the regression operator Γ lies inside the image of the log map centered in the dependent variable's barycenter, their distribution on distribution regression can be considered a properly intrinsic method. This assumption is used to prove asymptotic properties of their methodology, but as the authors in Chen et al. (2021) notice, is hardly verified in practice, so that whenever the output of the regression operator is not a distribution, they resort to squeezing such a value with some scalar multiplication, namely 'boundary projection', which in general is not a metric projection. The boundary projection step gives an extrinsic nature to their model and we provide further comparisons with our methods in Section 11.3.4.

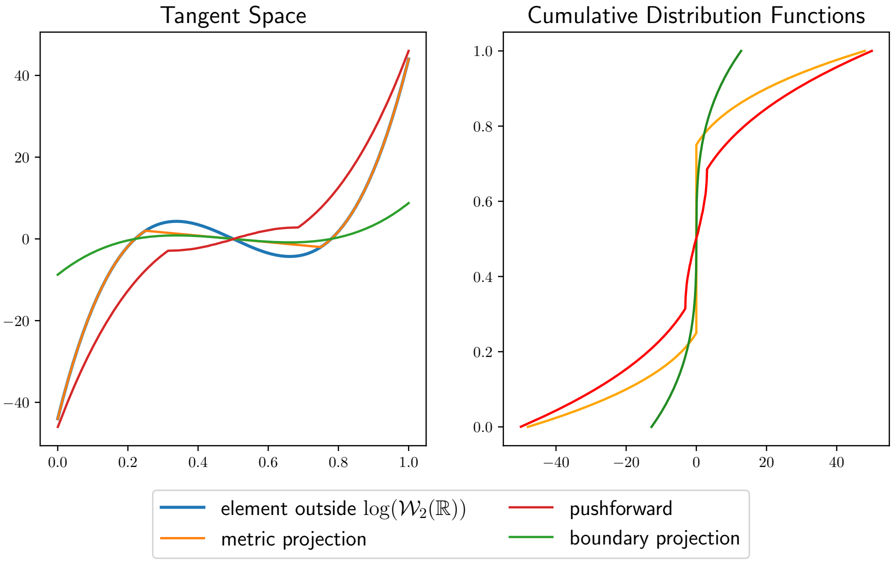


Figure 11.3.2: Comparison between different projections onto X for a point $x \in H \backslash X$ (blue line) in the tangent space (left panel) and the associated cumulative distribution functions (right panel) when the base point μ is the uniform measure on $[0, 1]$. The orange, green and red curves are obtained with metric projection, boundary projection and $\log_\mu \circ \exp_\mu$ respectively.

11.3.4 Comparison with other extrinsic methods

In this section, we offer a comparison of our projected methods with other extrinsic methods, namely the log PCA in Cazelles et al. (2018) and the distribution on distribution regression in Chen et al. (2021), which, as outlined in the previous section, may behave as an extrinsic method. Let us start with the former.

Cazelles et al. (2018) propose the definition of a log PCA as an alternative to the geodesic PCAs in Bigot et al. (2017). Both the log and the projected PCA are extrinsic methods: they proceed by carrying out the PCA in a linear space H and then map back the results to the Wasserstein space, following an approach which had already been proposed by Fletcher et al. (2004).

For the log PCA, H is the tangent space at μ, for the projected H is $L_2^\mu(\mathbb{R})$. Given $U_k = \{w_1, \ldots, w_k\}$ the first k H-principal components, the log principal component in $\mathcal{W}_2(\mathbb{R})$ is $\exp_\mu(Sp(U_k))$. Analogously, by considering the convex cone $X =: \log_\mu(\mathcal{W}_2(\mathbb{R})) \subseteq H$, the principal component in X is $\log_\mu\left(\exp_\mu(Sp(U_k))\right)$.

We note two key differences between the log and projected PCA. First, as pointed out in Remark 11.2, $\log_\mu \circ \exp_\mu$ is not a metric projection in L_2^μ so that given a point $x \in H \backslash X$, $\log_\mu(\exp_\mu(x))$ might end up being quite different from x. See for instance Figure 11.3.2 where for a point x (blue line) that is close (in the L_2^μ norm) to X, $\log_\mu(\exp_\mu(x))$ turns out to be rather far from x. In the context of PCA, this means that as soon as the projection onto $Sp(U_k)$ of observation lies outside of X the log PCA quickly loses its interpretability. Second, as discussed in Remark 11.4, there is no guarantee that $\log_\mu\left(\exp_\mu(Sp(U_k))\right)$ is contained in $Sp(U_k)$, its dimension might increase and it might not even be convex. For this same reason, in general, log PCA cannot define a set of (orthogonal) principal directions which span the principal component. Hence, it is not possible to work directly on the scores of the PCA.

Combined, we believe that the above-mentioned issues present a major drawback of the log PCA when compared to the projected PCA, as they prevent the possibility of performing proper dimensionality reduction and working on the scores of data points on the principal components. Finally, we also point out that approximating the $\exp_\mu$ map is a nontrivial task, involving computing numerically the preimages of an arbitrary large number of sets and numerical differentiation, that can lead to numerical instability of the log PCA.

We end this discussion with a comparison between the boundary projection in Chen et al. (2021) and the metric projection. Their difference, for a possible regression output $x \in H \backslash X$ is depicted in Figure 11.3.2. Note that, by construction, such a procedure shrinks the tails of the output. Even when the regression output is slightly outside the image of the log map, the boundary projection result can be extremely far from the regression output and from the metric projection in terms of Wasserstein distance. For example, in Figure 11.3.2, the regression output and the projected method assign positive probability to values in the range $[-45, 45]$, while the output of the boundary projection assigns zero probability to values outside $[-17, 17]$. This underrepresentation of the variability might be a crucial issue depending on the application considered.

11.4 Computing the metric projection through B-spline approximation

The projected methods defined in Section 11.3 depend heavily on the availability of projection operators on the closed convex cone $X = \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$. Being X a cone inside a linear space, such operators are always well defined, but their implementation might be nontrivial. In this section, we present a possible solution to this problem, based on choosing a particular μ as base point and constructing a B-spline representation of the cone X.

11.4.1 Choosing μ as the uniform distribution on $[0, 1]$

As already mentioned, our projected methods can be carried out by choosing μ arbitrarily and there is no theoretical difference between different choices of μ, cf. Section 11.2.2. Nonetheless, in practice, a clever choice of μ can lead to substantially easier and more numerically stable algorithms. For instance, by choosing a measure μ with compact support C in $\mathbb{R}$, then the ambient space becomes $L_2^\mu(C)$ since we work up to zero-measure sets. This greatly simplifies any numerical procedure since we could work with grids over bounded sets and do not need to resort to any truncation procedure, which would be mandatory in case the support of μ was unbounded. Moreover, note that evaluating the maps φ_μ in a certain measure ν amounts to computing the transport map $T_\mu^\nu = F_\nu^- \circ F_\mu$, hence it is clear that the choice of F_μ numerically influences the results.

For the aforementioned reasons, we argue that a reasonable choice is to center our analysis in $\mu = U([0, 1])$. In fact, in this case, $L_2^\mu(\mathbb{R}) = L_2([0, 1])$, and $F_\mu = id_{[0,1]}$ (the transport maps are simply given by quantile functions).

11.4.2 Metric Projection

Having chosen μ as Section 11.4.1 leads to an explicit characterization of the image of φ_μ as the set of square-integrable a.e. non-decreasing functions on $[0, 1]$. Hence, the operator Π_X in Section 11.3 is the metric projection onto the cone of a.e. non-decreasing functions in $L_2([0, 1])$.

Projection onto monotone functions has been widely studied in the field of *order restricted* inference, (Anevski et al., 2006; Dykstra et al., 2012). For instance, in Anevski and Soulier (2011) an explicit characterization of such a projection is given, which, however, does not lead to a closed-form solution, while in Ayer et al. (1955) several numerical algorithms to approximate the projection operator are proposed. Those algorithms are based on approximating the function to be projected with a step function defined on n intervals and can be shown to have a computational complexity that is linear in n (Best and Chakravarti, 1990).

Despite the numerical convenience of the aforementioned approximations, we believe that they are not suited for distributional data analysis. First and foremost, suppose that observations are given as probability density functions, so that one may want to interpret the results of a PCA, for instance, in terms of pdfs and not of quantile functions. If one were to estimate discontinuous principal directions through any of the algorithms in Ayer et al. (1955), it would not be possible to do so, as the corresponding cdfs would not be differentiable. In addition to that, the choice of the number of intervals n is not obvious when quantile functions are not directly observed but obtained with transformation. If n needs to be big to faithfully approximate the true quantile functions, this projection can be quite slow.

For these reasons, we propose a B-spline expansion through which we can derive an alternative approximation of the projection operator Π_X, without incurring in the issues of the algorithms in Ayer et al. (1955). Moreover, we will also show in Section 11.5.3 that the proposed B-spline expansion also leads us to a simpler and faster reformulation of the geodesic PCA in Bigot et al. (2017).

11.4.3 Monotone B-splines representation

In what follows, let $\mu = U([0,1])$. Moreover, denote with $\boldsymbol{x} = [x_1, \dots, x_k]' \in \mathbb{R}^k$ a generic vector.

As already said, through the φ_μ map, we can identify $\mathcal{W}_2(\mathbb{R})$ with the space

$$L_2([0,1])^{\uparrow} := \{F^- \in L_2([0,1]) \text{ s.t. } F^- \text{ is monotonically nondecreasing}\}$$

This leads us to consider a suitable B-spline basis for the space to efficiently evaluate all the computations needed in our algorithms and for a convenient way to express the constraints which define $L^2([0,1])^{\uparrow}$. In particular, we consider the basis of quadratic splines with equispaced knots in $[0,1]$. The reason for this particular choice is two-folded. First of all, splines of degree greater than one enjoy the nice property of uniform approximation of all continuous functions as the maximum distance between knots goes to zero. In turn, this means that the closure of the linear space generated by the spline basis w.r.t the L_2 norm coincides with $L_2([0,1])$. Secondly, quadratic splines are particularly well suited to characterize monotonic functions by looking at the coefficients of the (quadratic) B-spline expansion, as shown in the next proposition.

Proposition 11.4. *Let $\{\psi_j^k\}_{j=1}^J$ be a basis of B-splines of order k defined over the knots $x_1, \dots, x_{J+k+2}$. Let $f(x) = \sum_{j=1}^J a_j \psi_j^k(x)$, then:*

1. *If the coefficients $\{a_j\}$ are monotonically increasing (decreasing) f is monotonically increasing (decreasing).*

2. *If $k = 2$, then 1. holds with an 'if and only if'.*

Before proceeding, let us fix some notation. From now on, we omit the dimension index 'k' for the spline basis, writing ψ_j for ψ_j^2, moreover we will let $\{\psi_j\}_{j=1}^J$ with fixed $J > 0$ denote a B-spline basis in $L_2([0,1])$.

Remark 11.7. *Let $\mathbb{R}^{J\uparrow}$ be the set of vectors $v \in \mathbb{R}^J$ with nondecreasing coefficients. That is, letting $G = \{g_{ij}\}$ be the $J \times J$ binary matrix such that $\sum_j g_{ij} v_j = v_i - v_{i-1}$, for any* *element $\boldsymbol{v} \in \mathbb{R}^J$ it holds that $G\boldsymbol{v} \geq 0$. Using Proposition 11.4, through the coordinates operator, the set $L_2([0,1])^{\uparrow} \cap Span\{\psi_j\}_{j=1}^J$ is fully identifiable with $\mathbb{R}^{J\uparrow}$, endowed with the metric given by the symmetric positive definite matrix E with entries*

$$E_{ij} = \langle \psi_i, \psi_j \rangle_{L_2([0,1])}. \tag{11.9}$$

The norm induced is therefore $\|\boldsymbol{x}\|_E^2 = \boldsymbol{x}^T E \boldsymbol{x}$.

Remark 11.8. *It is possible to find a basis for $\mathbb{R}^J$ with vectors lying in $\mathbb{R}^{J\uparrow}$ (and so in X_J), namely the vectors $(0,\ldots,0,1)$, $(0,\ldots,0,1,1)$ etc. In other words, $Span(L_2([0,1])^{\uparrow} \cap Span\{\psi_j\}_{j=1}^J) = Span\{\psi_j\}_{j=1}^J$ for every $J > 0$. This tells us that the convex cone of monotone splines is indeed quite big inside the spline space, and this a priori is beneficial for extrinsic methods, especially for PCA.*

From now on, to lighten the notation, we deliberately confuse the coefficients of the splines, living in $\mathbb{R}^J$ or $\mathbb{R}^{J\uparrow}$ (with the metric given by E), with the corresponding spline functions living in the subsets of $L_2([0,1])$ given by $L_2([0,1])^{\uparrow} \cap Span\{\psi_j\}_{j=1}^J$ and $Span\{\psi_j\}_{j=1}^J$.

Remark 11.9. *Lastly, we point out that $\mathbb{R}^{J\uparrow}$ has the structure of a convex polytope, since the constraints given by $G\boldsymbol{v} \geq 0$ (guaranteeing that $\boldsymbol{v} \in \mathbb{R}^{J\uparrow}$) are linear. Such geometric property makes optimization on $\mathbb{R}^{J\uparrow}$ handy and is key for the empirical methods developed in the remaining of the chapter.*

As a consequence of Remark 11.9, the optimization problem given by the projection of a vector $\boldsymbol{v} \in \mathbb{R}^J$ onto $\mathbb{R}^{J\uparrow}$ can be formulated as follows:

$$\Pi_{\mathbb{R}^{J\uparrow}}(\boldsymbol{v}) = \operatorname*{arg\,min}_{G\boldsymbol{w} \geq 0} \|v - w\|_E. \tag{11.10}$$

The computational complexity required to solve (11.10) is at most cubic in the number of basis elements J (Potra and Wright, 2000).

Preliminary analysis showed that solving the optimization problem in (11.10) compares favorably with the Pool Adjacent Violators Algorithm (PAVA) in Ayer et al. (1955). In particular, computing PAVA with $n = 100$ approximation intervals is roughly eight times slower than (11.10) with $J = 20$ (a reasonable choice, leading to negligible approximation error, in our examples, with a quadratic spline basis). Increasing $n = 1000$ for PAVA makes it 700 times slower than (11.10).

In addition to that, resorting to a discretized approximation of quantiles would also increase the cost of the projected PCA, due to the need of using some functional PCA implementation, as opposed to the low-dimensional multivariate model we are able to implement with the B-spline basis functions.

11.5 Empirical Models with B-splines

In this section, we present the empirical counterparts of the projected PCA defined in Section 11.3 and provide an illustrative example of projected linear regression, namely when both the dependent and independent variables are distributions.

Let $\{\psi_j\}_{j=1}^J$ be a fixed quadratic B-spline basis. Upon approximating the observed quantile functions with their spline expansion, thanks to Remark 11.7, we can develop our methodology in $\mathbb{R}^J$, considering the metric induced by E instead of the usual one. Indeed, given a vector $\boldsymbol{w} \in \mathbb{R}^J$, we can identify the corresponding function in L_2 by the map $\boldsymbol{w} \mapsto \sum_{j=}^J w_j \psi_j$.

For the projected PCA in Section 11.5.1 and for the geodesic PCA in Section 11.5.3 we consider observations $F_1^-, \ldots, F_n^-$ and let F_0^- be the centering point of the PCA. In our examples, F_0^- will always be the barycenter of the observations. As a preprocessing step, we approximate each of these quantile functions through a B-spline expansion and denote by $\boldsymbol{a}_i = \{a_{ij}\}_j$ and $\boldsymbol{a}_0 = \{a_{0j}\}_j$ the coefficients of the spline representation associated to F_i^- and F_0^- respectively, that is, $F_i^- \approx \sum_{j=}^J a_{ij}\psi_j$. For the projected regression in Section 11.5.2, let observations $\{(F_z^-, F_y^-)_i\}_{i=1}^n$, where the F_{zi}^-'s are realizations of the independent variable $\mathcal{Z}$ and the F_{yi}^-'s are realizations of the dependent variable $\mathcal{Y}$. We apply the same preprocessing step and let $\boldsymbol{a}_i^{(z)}$ and $\boldsymbol{a}_i^{(y)}$ denote the coefficient of the spline approximation of F_{zi}^- and F_{yi}^- respectively.

11.5.1 Empirical PCA

As in standard PCA, the first principal component centered in $\boldsymbol{a}_0$ is found by solving the optimization problem:

$$\boldsymbol{w}_1^* = \underset{\boldsymbol{w}:\|w\|_E=1}{\arg\max} \sum_i |\langle \boldsymbol{a}_i - \boldsymbol{a}_0, \boldsymbol{w}\rangle_E|^2 = \underset{\boldsymbol{w}:\|\boldsymbol{w}\|_E=1}{\arg\max} \|AE\boldsymbol{w}\|^2, \tag{11.11}$$

where A is the $n \times J$ matrix whose i–th row is given by $\boldsymbol{a}_i - \boldsymbol{a}_0$. The optimization problem (11.11) can be solved similarly to a Rayleigh quotient: using Lagrange multipliers, (11.11) is equivalent to

$$\mathcal{L}(\boldsymbol{w}) := \boldsymbol{w}^T (A\ E)^T\ A\ E\ \boldsymbol{w} - \lambda(\boldsymbol{w}^T\ E\ \boldsymbol{w} - 1). \tag{11.12}$$

Deriving (11.12) w.r.t $\boldsymbol{w}$ and equating the derivative to zero shows that the solutions to $d\mathcal{L}(\boldsymbol{w})/d\boldsymbol{w} = 0$ are the eigenvectors of the matrix A^TAE. Hence, ordering the eigenvalues of A^TAE in decreasing order, the first principal component $\boldsymbol{w}_1^*$ corresponds to the first eigenvector. Using similar arguments, it can be shown that $\boldsymbol{w}_2^*, \dots \boldsymbol{w}_J^*$ correspond to the remaining eigenvectors.

Once the first k principal directions $\boldsymbol{w}_1^*, \dots, \boldsymbol{w}_k^*$ are found, the projection of a new observation $x^* = \sum_{j=1}^J a_j^* \psi_j$ onto U_X^{k,x_0} (see Definition 1) is found exploiting Proposition 11.2. In particular, the following optimization problem is to be solved:

$$\begin{aligned} &\underset{\lambda_j \in \mathbb{R}}{\arg\min} \|(\langle \boldsymbol{a}^* - \boldsymbol{a}_0, \boldsymbol{w}_i^*\rangle_E - \lambda_i)_{i=1}^k\|, \\ &\text{s.t. } G\Big(\sum_{i=1}^k \lambda_i \boldsymbol{w}_i^* + \boldsymbol{a}_0\Big) \geq 0. \end{aligned} \tag{11.13}$$

which is equivalent to the minimization of a norm inside a polytope, that is a well-studied problem in $\mathbb{R}^J$ (see Sekitani and Yamamoto, 1993) and there exist a variety of fast numerical routines to solve it.

11.5.2 Empirical Regression

In this section, we provide the details of the estimation procedure for a projected regression model where both the independent and the dependent variables are distribution-valued. It is straightforward to extend our methodology to cases when only one of these variables is distribution-valued and the other one takes values in $\mathbb{R}^q$.

First, we outline how to obtain an estimator for the linear operator Γ in Definition 4. Following Section 11.3.2 we first embed both $\mathcal{Y}$ and $\mathcal{Z}$ in $L_2([0,1])$ through the inclusion operator $L_2([0,1])^\uparrow \hookrightarrow L_2([0,1])$, and assume the functional linear model presented in Ramsay (2004) and Prchal and Sarda (2007)

$$\mathcal{Y}(t) = \alpha(t) + \int_0^1 \beta(t,s)\mathcal{Z}(s)ds + \varepsilon(t), \qquad t \in [0,1], \tag{11.14}$$

so that $\Gamma = \Gamma_{\alpha,\beta}$ is the operator $\Gamma_{\alpha,\beta}(v)(t) = \alpha(t) + \int_0^1 \beta(t,s)v(s)ds$. The goal is then to estimate $\alpha \in L_2([0,1])$ and $\beta \in L_2([0,1]^2)$. Further, we assume that ε and $\mathcal{Z}$ are uncorrelated: $\mathbb{E}[\mathcal{Z}(s)\varepsilon(t)] = 0$ for every $t, s \in [0,1]$.

Consider now observations $\{(F_z^-, F_y^-)_i\}_{i=1}^n$ and the corresponding spline coefficients. Further, we project $\alpha(t)$ on the same spline basis, so that $\alpha \approx \sum_{j=1}^J \theta_{\alpha j}\psi(j)$ and $\beta(t,s)$ on the basis on $[0,1]^2$ with $J \times J$ elements, so that $\beta(t,s) \approx \sum_{i,j\prime=1}^J \Theta_{\beta ij}\psi_i(t)\psi_j(s)$. Neglecting the spline approximation error, model (11.14) entails

$$\boldsymbol{a}_i^{(y)} = \boldsymbol{\theta}_\alpha + \Theta_\beta E \boldsymbol{a}_i^{(z)} + \boldsymbol{a}_i^{(\varepsilon)}, \qquad i = 1, \dots, n, \tag{11.15}$$

where $\boldsymbol{a}_i^{(\varepsilon)}$ denotes the spline expansion coefficients of the unobserved error $\varepsilon_i(t)$.

We propose to estimate (11.15) using the same approach of Prchal and Sarda (2007), but extending it to account for spline approximations for both dependent and independent variables. We focus only on the estimate $\widehat{\Theta}_\beta$ of Θ_β since once such estimate is obtained, the estimate for $\boldsymbol{a}_\alpha$ can be straightforwardly derived (see Cai and Hall, 2006) as:

$$\hat{\boldsymbol{\theta}}_\alpha = \overline{\boldsymbol{a}^{(y)}} - \widehat{\Theta}_\beta E \overline{\boldsymbol{a}^{(z)}},$$

where $\overline{\boldsymbol{a}^{(y)}}$ and $\overline{\boldsymbol{a}^{(z)}}$ are the means of $\boldsymbol{a}^{(y)}$ and $\boldsymbol{a}^{(z)}$ respectively.

The estimator $\widehat{\Theta}_\beta$ is found by penalized least square minimization:

$$\widehat{\Theta}_\beta = \arg\min_{\Theta} \frac{1}{n}\sum_{i=1}^n \| \left(\boldsymbol{a}_i^{(y)} - \overline{\boldsymbol{a}^{(y)}}\right) - \Theta E \left(\boldsymbol{a}_i^{(z)} - \overline{\boldsymbol{a}^{(z)}}\right) \|^2 + \rho \mathrm{Pen}(1, \Theta), \tag{11.16}$$

where $\rho > 0$ is a penalization parameter to be fixed (usually through cross-validation) and $\mathrm{Pen}(1, \Theta)$ is a penalization term defined in Prchal and Sarda (2007).

Briefly, the term $\mathrm{Pen}(1, \Theta)$ in (11.16) penalizes both the norm of $\beta(t,s)$ and its derivatives, thus favoring smoother solutions. As shown in Prchal and Sarda (2007), (11.16) has a closed form solution. Nonetheless, the form of our solution differs from the one presented in Prchal and Sarda (2007), since they work directly on discretized functions while we propose to estimate spline coefficients and some care must be taken since they can use (up to scaling) the usual inner product in the Euclidean space of discretized functions, while we must consider the inner product induced by E. However, the procedure for obtaining our result is identical to the one in Prchal and Sarda (2007). Hence, we only report the expression for the estimate.

Let $\hat{C}$ be the matrix with entries

$$\hat{C}_{ks} = \left\langle \frac{1}{n}\sum_{i=1}^n \langle \boldsymbol{a}_i^{(z)}, b_k\rangle_E \ \boldsymbol{a}_i^{(z)}, b_s \right\rangle_E,$$

where b_k and b_s are the k-th and s-th elements of the standard Euclidean basis in $\mathbb{R}^J$. Further, let $\hat{D}$ the matrix with entries

$$\hat{D}_{ks} = \left\langle \frac{1}{n}\sum_{i=1}^n \langle \boldsymbol{a}_i^{(z)}, b_k\rangle_E \ \boldsymbol{a}_i^{(y)}, b_s \right\rangle_E.$$

Finally, let E' denote the matrix with entries $E'_{ij} =< \psi'_i, \psi'_j >$ (where ψ'_i denotes the first derivative of the B-spline basis function ψ_i), $C_\rho = E^T \otimes (\hat{C} + \rho E')$, and $P = E'^T \otimes E + E^T \otimes E'$, where $\otimes$ denotes the Kronecker product. Then the solution of (11.16) can be expressed as

$$\mathrm{vec}(\widehat{\Theta}_\beta) = (C_\rho + \rho P)^{-1}\mathrm{vec}(\hat{D}),$$

where vec$(\cdot)$ denotes the *vectorization* of the matrix.

Finally, our projected regression model is the composition of the operator induced by $(\hat{\boldsymbol{\theta}}_\alpha, \hat{\Theta}_\beta)$ with the projection on $\mathbb{R}^{\uparrow J}$:

$$\mathbb{E}[\boldsymbol{a}_i^{(y)} \mid \boldsymbol{a}_i^{(z)}] = \Gamma_{\mathrm{P}}(\boldsymbol{a}_i^{(z)}) = \Pi_{\mathbb{R}^{J\uparrow}} \left(\hat{\boldsymbol{\theta}}_\alpha + \widehat{\Theta}_\beta E \boldsymbol{a}_i^{(z)} \right).$$

11.5.3 An alternative optimization routine for the geodesic PCA and a comment on the computational costs

We now show how the framework in Section 11.4 can be employed also to derive faster numerical algorithms to find the global and nested geodesic PCA as of Definition 5 and Definition 6.

Proposition 11.5. *(Global geodesic PCA) A k dimensional global geodesic PC centered in $\boldsymbol{a}_0$ is the subset of $\mathbb{R}^{J\uparrow}$ spanned by $\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_k\}$, linearly independent, which solve:*

$$\begin{aligned} &\underset{\{\boldsymbol{\lambda}_i\}_1^n, \{\boldsymbol{w}_j\}_1^k}{\arg\min} \sum_{i=1}^n ||\boldsymbol{a_i} - \boldsymbol{a_0} - \sum_{j=1}^k \lambda_{ij} \cdot \boldsymbol{w}_j||_E^2, \\ &\quad s.t.\ G\Big(\sum_j \lambda_{ij} \boldsymbol{w}_j + \boldsymbol{a}_0 \Big) \geq 0. \end{aligned} \tag{11.17}$$

Proposition 11.6. *(Nested geodesic PCA) With the same notation as above, a k dimensional nested geodesic PC, centered in $\boldsymbol{a}_0$ is the set spanned by $\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_k\}$ in $\mathbb{R}^{J\uparrow}$, where the $\boldsymbol{w}_i$s are found recursively from $\boldsymbol{w}_1$ to $\boldsymbol{w}_k$, such that $\boldsymbol{w}_h$ is a solution, for every h, of:*

$$\begin{aligned} &\underset{\{\boldsymbol{\lambda}_i\}_{i=1}^n, \boldsymbol{w}}{\arg\min} \sum_{i=1}^n \|\boldsymbol{a_i} - \boldsymbol{a_0} - \lambda_i \boldsymbol{w}\|_E^2, \\ &\quad s.t.\ \langle \boldsymbol{w}_j, \boldsymbol{w} \rangle_E = 0, \quad j = 1, \dots, h-1, \\ &\quad G\Big(\lambda_i \boldsymbol{w} + \boldsymbol{a_0} \Big) \geq 0, \quad \|\boldsymbol{w}\|_E = 1. \end{aligned} \tag{11.18}$$

To solve (11.17) and (11.18) we employ an interior point method using the solver Ipopt (Waechter and Biegler, 2006). When comparing our implementation with $J = 20$ spline basis and the one in Cazelles et al. (2018), we notice a substantial performance improvement, by a factor of 35 for a data set of $n = 100$ distributions, due to the fact that working with spline approximations greatly reduces the number of parameters in the optimization problem.

Further, note that (11.17) and (11.8) seem extremely similar. However, in (11.8) the optimization is carried out having fixed $\boldsymbol{w}_1^*, \dots, \boldsymbol{w}_k^*$ and for a single observation, while in (11.17) the optimization is done over a much larger set of parameters. In fact, the number of parameters in (11.17) is $(n + k)J$, hence the computational complexity needed to solve (11.17) is cubic in both the number of bases and the number of observations. On the other hand, the projected PCA requires a linear time in the number of observations (computation of $A^T AE$) and cubic time in the number of basis J (eigendecomposition and projections of new observations).

11.6 Asymptotic Properties

In this section, we study the convergence of the proposed projected empirical methods. First of all, we show that as the number of spline basis J increases, the error due to the spline approximation vanishes if the data is sufficiently regular. Further, under a suitable set of assumptions, we establish consistency results for the projected PCA and for the projected distribution on distribution regression.

11.6.1 Convergence of Quadratic B-splines

In the following, denote with $W_k^r([0,1])$ the Sobolev space of functions whose weak derivatives up to order k belong to $L_r([0,1])$, further denote with D the (weak) derivative operator, so that $Df = f'$, $D^2 f = f''$ and so on,

Proposition 11.7. *Let μ a probability measure on $\mathbb{R}$, F_μ^- its quantile function such that $F_\mu^- \in W_3^\infty$. For each J let $\{\psi_j\}_{j=1}^J$ denote a quadratic B-spline basis on J equispaced knots in $[0,1]$. Then there exist a sequence of spline functions $S_J = \sum_{j=1}^J \lambda^{(J)} \psi_j^{(J)}$, with $\lambda_j^{(J)}$ monotonically non-decreasing in j for every J, such that:*

$$\|S_J - F_\mu^-\|_\infty \leq C \|D^2 f_\mu^-\|_\infty J^{-2},$$

with $f_\mu^- = DF_\mu^-$ and $C > 0$ constant.

Let us remark two important facts.

Remark 11.10. *Since the inclusion $L_\infty([0,1]) \subset L_2([0,1])$ is continuous, thanks to Hölder inequality, the convergence rates hold also for the L_2 norm. By default we will use the L_2 norm if not stated differently.*

Remark 11.11. *By Poincaré inequality, if $\|D^3 f\|_\infty < C$ then f belongs to a sphere in $W_3^\infty([0,1])$ whose radius depends on C and on the Poincaré constant of $[0,1]$; viceversa, all the elements in the sphere of radius C in $W_3^\infty([0,1])$ clearly have (weak) derivatives bounded by C.*

11.6.2 Consistency

In this section we prove the consistency of the projected methods under some assumptions on the data-generating process. In particular, we show that that there exists a number of basis functions $J > 0$ and a sample size n such that the error committed by the empirical models in Section 11.5 is smaller than $\varepsilon > 0$, for any fixed ε.

11.6.2.1 PCA

The consistency of spline-based PCA for functional data has been addressed, among the first, by Silverman et al. (1996) and Qi and Zhao (2011). As one of the main building blocks of our projected PCA is the PCA in the ambient space, that is $L_2([0,1])$, it is natural to follow Qi and Zhao (2011) in making the following assumptions. Consider data $\mu_1, \ldots, \mu_n$, $F_1^-, \ldots, F_n^-$ the corresponding quantile functions, then:

(P1) The data generating process satisfies $F_1^-, \ldots, F_n^- \sim \mathcal{F}$ with the F_i^- independent and $\mathbb{E}[\mathcal{F}] = 0$.

(P2) $F_1^-, \ldots, F_n^-$ can be approximated by functions in W_3^∞ with uniformly bounded third derivative.

(P3) $\mathbb{E}[\|F_i^-(t)\|^4] < \infty,\ i = 1, \ldots, n$.

(P4) The eigenvalues of the covariance operator of $\mathcal{F}$ have multiplicity 1.

(P5) The eigenfunctions of the covariance operator of $\mathcal{F}$ belong to some bounded set in $W_3^\infty([0,1]) \subset W_3^2([0,1])$.

Before stating the main results, let us comment on assumptions (P1)-(P5). First of all, (P2) is essential in order to apply Proposition 11.7 and get uniform errors on the data set. Moreover, (P2) is satisfied, for instance, if the F_i^-'s lie in the L_2-closure of a ball of radius $M > 0$ in W_3^∞. (P4) is a rather standard condition and is satisfied if $\mu_1, \dots, \mu_n \in \mathcal{W}_4(\mathbb{R})$. (P4) and (P5) imply the assumptions that in Qi and Zhao (2011) are used for the consistency results. In particular, (P5) is stronger than the corresponding assumption in Qi and Zhao (2011), where the eigenfunctions are assumed to belong to $W_2^2([a, b])$. Similarly, in such work, there is no counterpart of assumption (P2); in fact, we need these stronger regularity conditions to get uniform errors when using B-splines. Still, some of the examples Qi and Zhao (2011) provide of situations satisfying their assumptions meet also our requirements. Finally, the zero-mean assumption in (P1) might seem a little odd, since we know that the quantile functions are monotonically nondecreasing. However, observe that it is always possible to subtract the empirical mean from the observations to satisfy (asymptotically) this assumption.

Let J denote the dimension of a quadratic B-spline basis on $[0, 1]$ and let $\boldsymbol{a}_i^J$ the coefficients of the B-spline approximation of F_i^-. In what follows, to lighten the notation, we refer to a set of spline coefficients both as elements of $\mathbb{R}^J$ with the E-norm, or as functions in L_2, without making explicit reference to the coordinate operator and its inverse.

Proposition 11.8. *Under assumptions (P1)-(P5), for any $\varepsilon > 0$ there exists a sample size $n > 0$ and a number of basis functions $J > 0$ such that:*

$$\left| \max_{\|w\|_{L_2}=1} \frac{1}{n} \sum_i \langle F_i^-, w\rangle_{L_2}^2 - \max_{\|\boldsymbol{w}\|_E=1} \frac{1}{n} \sum_i \langle \boldsymbol{a}_i^J, \boldsymbol{w}\rangle_E^2 \right| < K\varepsilon,$$

for some constant $K > 0$.

Proposition 11.8 ensures the consistency of the B-spline approximation of the PCA for monotone functional data in H, which is equivalent to the consistent estimation of the projected principal directions.

Suppose now to have computed $U_k^J = \{\boldsymbol{w}_h^{J*}\}_{h=1}^k$, that is the approximations of the principal directions $U_k = \{\boldsymbol{w}_h^*\}_{h=1}^k$ found with J basis functions. We observe that $Sp(U_k^J) \cap L_2([0, 1])^\uparrow = Sp(U_k^J) \cap \mathbb{R}^{J\uparrow}$. Since for any set of coefficients λ_h we have the convergence $\sum \lambda_h \boldsymbol{w}_h^{J*} \to \sum \lambda_h w_h^*$, we obtain that the projection of a point onto $Sp(U_k^J) \cap L_2([0, 1])^\uparrow$ converges to the projection onto $Sp(U_k) \cap L_2([0, 1])^\uparrow$. Thus we also have convergence of the projection onto the principal components.

11.6.2.2 Regression

We consider model (11.14) given samples $\{(F_z^-, F_y^-)_i\}_{i=1}^n$. We make the following assumptions:

(R1) The data generating process satisfies (11.14) and $\mathbb{E}[\mathcal{Z}(s)\varepsilon(t)] = 0$ for every $t, s \in [0, 1]$.

(R2) $\alpha \in L_2([0, 1])$ and $\beta \in L_2([0, 1] \times [0, 1])$.

(R3) With probability 1, each quantile function in the samples $\{(F_z^-, F_y^-)_i\}_{i=1}^n$ lies inside a sphere of radius $K > 0$ in $W_\infty^3([0, 1])$.

Without loss of generality, suppose that both the dependent and the independent variables have been centered by subtracting their mean so that $\mathbb{E}[\mathcal{Z}] = \mathbb{E}[\mathcal{Y}] = 0$ and $\alpha = 0$.

The strategy to prove the consistency of the projected linear regression is the following. First of all, we prove that the estimator $\widehat{\Theta}_J$ converges to the estimator $\widehat{\Theta}_{\text{PS}}$, defined in Prchal and Sarda (2007), for large enough n and J. Second, we exploit the consistency of the estimator in Prchal and Sarda (2007) combined with the approximation results of the metric projection to establish consistency in terms of the prediction error of our projected regression operator.

Briefly $\widehat{\Theta}_{\text{PS}}$ is obtained by minimizing an objective function similar to the one in (11.16), but where the spline approximation is used only for Θ, while the F^{-}_{zi}'s and the F^{-}_{yi}'s are assumed fully observed and not approximated through splines. Calling B the vector of functions with entries $\psi_1, \dots, \psi_J$, $\widehat{\Theta}_{\text{PS}}$ is defined as:

$$\widehat{\Theta}_{\text{PS}} = \arg\min_{\Theta} \frac{1}{n} \sum_i \|F^{-}_{yi} - \langle F^{-}_{zi}, B^T \Theta B\rangle\|^2 + \rho \text{Pen}(1, \Theta).$$

Convergence of $\widehat{\Theta}_J$ to $\widehat{\Theta}_{\text{PS}}$ is shown in the next proposition

Proposition 11.9. *Under assumptions (R1)-(R3), if the number of samples is big enough, $\widehat{\Theta}$ and $\widehat{\Theta}_J$ exist with probability close to* 1 *and there is $J > 0$ such that $\|\widehat{\Theta}_{PS} - \widehat{\Theta}_J\|_{E\otimes E} < \varepsilon$.*

Let $\widehat{\beta}_{\text{PS}}$ and $\widehat{\beta}_J$ be the kernels $\widehat{\beta}_{\text{PS}} = B^T \widehat{\Theta}_{\text{PS}} B$ and $\widehat{\beta}_J = B^T \widehat{\Theta}_J B$. Since $\|\widehat{\beta}_{\text{PS}}(s,t) - \widehat{\beta}_J(s,t)\|_{L_2([0,1]^2)} = \|\widehat{\Theta}_{\text{PS}} - \widehat{\Theta}_J\|_{E\otimes E}$, we established strong convergence of our kernel to the estimator of Prchal and Sarda (2007). This implies that the consistency results for the estimator $\widehat{\Theta}_{\text{PS}}$ holds also for $\widehat{\Theta}_J$, with respect to the seminorm induced by the covariance operator of $\mathcal{Z}$. Specifically, given $\mathcal{Z}$, H-valued random variable, and its covariance operator $\mathcal{C}_{\mathcal{Z}}$, for any $\varphi \in L_2([0,1]^2)$, we consider the semi-norm on $L_2([0,1]^2)$ given by:

$$\|\varphi\|_{\Gamma_{\mathcal{Z}}} = \int_{[0,1]} \langle \mathcal{C}_{\mathcal{Z}} \varphi(\cdot, t), \varphi(\cdot, t)\rangle dt.$$

Thus, the following result is immediately implied since strong convergence implies seminorm convergence (see Appendix 11.A).

Corollary 11.1. *For $J > 0$ big enough $\mathbb{E}[\|\beta - \widehat{\beta}_J\|_{\mathcal{C}_{\mathcal{Z}}}] < \varepsilon$.*

Proof. We use the seminorm triangle inequality:

$$\|\beta - \widehat{\beta}_J\|_{\mathcal{C}_{\mathcal{Z}}} \leq \|\beta - \widehat{\beta}\|_{\mathcal{C}_{\mathcal{Z}}} + \|\widehat{\beta} - \widehat{\beta}_J\|_{\mathcal{C}_{\mathcal{Z}}}.$$

The first term on the right-hand side converges to zero thanks to Theorem 2 in Prchal and Sarda (2007), while the second term converges to zero thanks to Proposition 11.9 and the previous observations. □

Lastly, we need to take into account the projection step. First, we notice that $\|\beta - \widehat{\beta}\|_{\Gamma_{\mathcal{Z}}}$ corresponds to the expected prediction error, in fact, as in Prchal and Sarda (2007):

$$\|\beta - \widehat{\beta}_J\|_{\mathcal{C}_{\mathcal{Z}}} = \int_{[0,1]} \mathbb{E}\left[\langle \mathcal{Z}, \beta(\cdot,t) - \widehat{\beta}_J(\cdot,t)\rangle^2 \mid \widehat{\beta}_J\right] dt,$$

further, by Hölder's inequality $\mathbb{E}\left[|\langle \mathcal{Z}, \beta - \widehat{\beta}_J\rangle| \mid \widehat{\beta}_J\right] \to 0$, which straightforwardly yields $\mathbb{E}\left[\|\Gamma_\beta(z) - \Gamma_{\widehat{\beta}_J}(z)\| \mid \widehat{\beta}_J\right] \to 0$.

Thus, the following simple lemma ensures the consistency of the spline approximation of the projection on X and leads to the consistency of the projected regression in terms of prediction error. Again, following Remark 11.7, we can identify the space monotone B-splines with J basis functions with $\mathbb{R}^{J\uparrow}$. Hence, to lighten the notation, we denote $\Pi_{\mathbb{R}^{J\uparrow}}$ the metric projection operator onto the space of monotone B-splines with J basis functions.

Lemma 11.2. *Given $b_n \to b$ in H, for any $\varepsilon > 0$ there exists $n, J > 0$ such that $\|\Pi_{\mathbb{R}^{J\uparrow}}(b_n) - \Pi_{L_2([0,1])^{\uparrow}}(b)\| \leq \varepsilon$.*

11.7 Numerical Illustrations for the PCA

In this section, we perform PCA on different simulated data sets and on a real data set of Covid-19 mortality data in the US. In particular, on the simulated data sets, we compare the performance of our projected PCA (in terms of approximation error and interpretability of the directions) with the ones of intrinsic methods, showing that the projected PCA is a valid competitor in a diverse set of situations. For the Covid-19 data set, we compare inference obtained using the projected, nested and log PCA, highlighting the practical benefits of the projected PCA over the log one.

For the projected, nested, and global PCAs we need to fix a B-spline basis to express the quantile functions. In particular, we fix an equispaced quadratic B-spline basis with J interior knots on $[0, 1]$. Here, the number of basis J is always fixed to 20, which provided a negligible approximation error of the quantile functions. We did not observe any appreciable change when increasing it. In Appendix 11.C we show further simulations where we perform sensitivity analysis as the number of basis increases for a fixed sample size, we provide empirical confirmation of the consistency results in Section 11.6 and give practical guidance on how to choose J.

11.7.1 Simulation studies

We consider three different simulations to compare both the interpretability and the ability to compress information of different PCAs.

We compare our projected PCA with the nested and global geodesic PCAs (Bigot et al., 2017; Cazelles et al., 2018) and the *simplicial* PCA (Hron et al., 2014).

Briefly, the simplicial PCA applies a transformation that maps densities defined on the same compact interval I into functions in $L_2(I)$, called *centered log ratio*. Then, a standard L_2 PCA is performed on the transformed pdfs and, by the inverse of the centered log ratio transform, the results are mapped back to the space of densities, called Bayes space (for a more accurate definition, see Egozcue et al., 2006). In particular, we remark that, in order to be well defined, the simplicial PCA requires that all the pdfs have support equal to I, which is a strong assumption in practice. Further details about simplicial PCA are given in Appendix 11.B.

As for the projected PCA, to compute the simplicial PCA, we resort to a B-spline approximation, but this time of the transformed pdfs. Hence, we need to select a B-spline basis on the support of the pdfs I. In this case, we fix a cubic B-spline basis with $J' = J = 20$ interior knots on I, as this choice yielded a negligible approximation error for the transformed pdfs.

In the first scenario, we simulate data from

$$\begin{aligned} p_i(x) &\propto \frac{1}{\sigma_i} \exp\left((x-\mu_i)^2/(2\sigma_i^2)\right) \mathbb{I}(x \in [-10, 10]), \quad i = 1, \dots 100, \\ \mu_i &\sim 0.5\mathcal{N}(-3, (0.2)^2) + 0.5\mathcal{N}(3, (0.2)^2), \\ \sigma_i &\sim \text{Uniform}([0.5, 2.0]). \end{aligned} \tag{11.19}$$

Where 'proportional to' stands for the fact that we confine the density to the support $[-10, 10]$ and renormalize it so that it integrates to 1.

Observe that there are two sources of variability across the pdfs from the data generating process (11.19). The first one is the location of the *peak* μ_i and the second one is the *width* of the distribution around the peak, controlled by σ_i. See Figure 11.7.1.

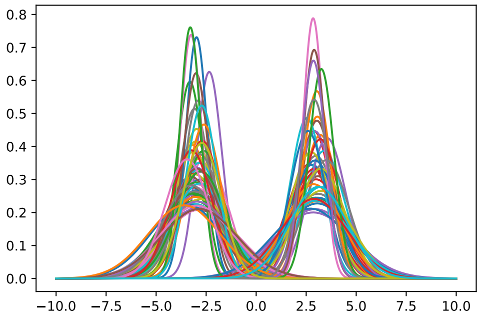


Figure 11.7.1: Data set of pdfs generated from (11.19)

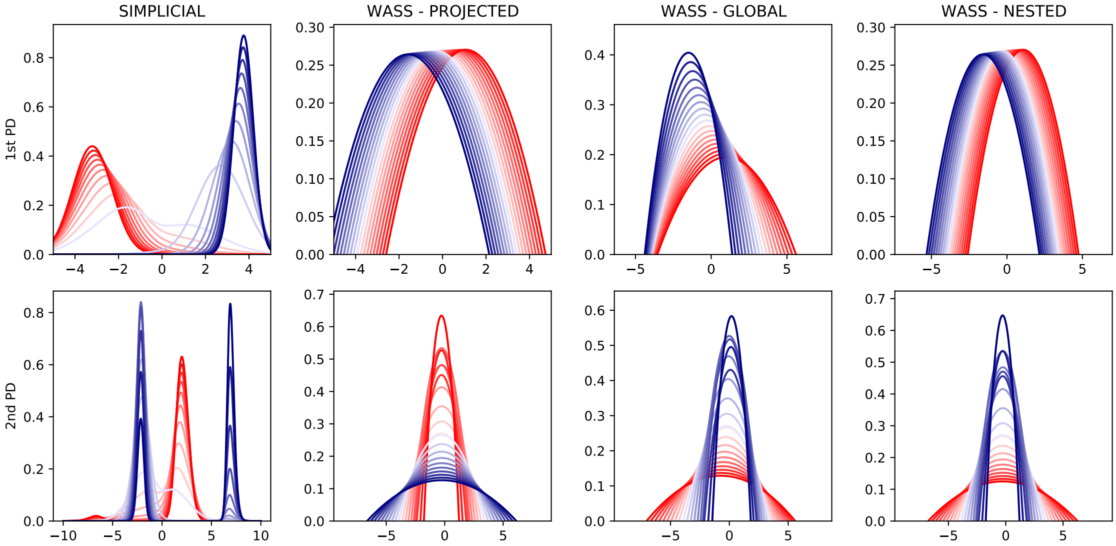


Figure 11.7.2: Top row: first principal direction. Bottom row: second principal direction. Each line represents the pdf associated to $\lambda \boldsymbol{w}_i$ where $\boldsymbol{w}_i$ is the i–th principal direction ($i = 1, 2$) and λ is a score ranging from -2 (darkest blue) to $+2$ (darkest red).

Figure 11.7.2 shows the first two principal directions obtained using the different methods. We can notice several differences between them. Focusing on the first principal direction, we can see that the simplicial, projected, and nested PCAs detect a change in the location of the peak of the pdf. In particular, the first direction for the Wasserstein PCAs represents a shift from left to right of this peak, while, for the simplicial PCA, the first direction is associated to a peak in 3 (blue lines, negative values of the scores) or to a peak in -3 (red lines, positive value of the scores). This also highlights the difference in the geometries underlying the Wasserstein and Bayes spaces. Looking at the second principal direction instead, we can see how in the Wasserstein PCAs it clearly represents a change in the width of the distribution, while for the simplicial PCA the interpretation is somewhat obscure.

The global geodesic PCA deserves a separate discussion. Indeed, from Definition 5 it is clear that a global principal component is a convex set without any notion of preferential directions, so that it is not possible to interpret separately the variation along the first and second direction found by the global PCA.

Now we present two additional simulations that quantify the amount of information that is 'lost' by performing the PCA. As a metric, we consider the reconstruction error, that is, the quantity

$$RE_k = \frac{1}{n}\sum_{i=1}^{n} W_2(F_i^-, \widetilde{F}_i^-), \tag{11.20}$$

where the F_i^-'s are the observed probability measures, $\widetilde{F}_i^-$ are the reconstructed ones and k is the dimension of the principal component. More in detail, $\widetilde{F}_i^-$ is found by first projecting $(F_i^- - F_0^-)$ into $\mathbb{R}^k$ using the PCA and then applying the inverse transformation. Informally, the reconstruction error is a measure of the quantity of information lost by applying the PCA as a black-box dimensionality reduction.

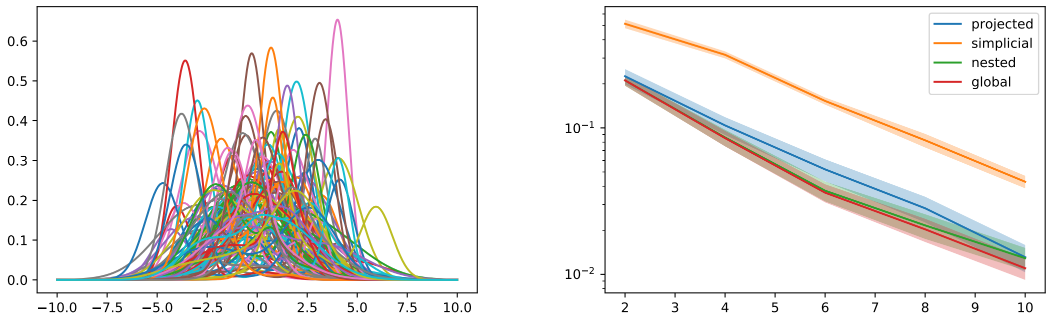
As evident in Equation (11.20), we measure the performance of PCAs just in terms of Wasserstein metric. This is likely to favor the performance of the Wasserstein PCAs over the simplicial one. Thus, the interesting performance comparison is the one between the geodesic PCAs and the projected PCA. Nevertheless, we think that is worth reporting also the results for the simplicial PCA, which is an intrinsic method in the Bayes space, to show that the underlying metric structures are extremely different. This also helps to appreciate the results in Section 11.8. Given the difference in the metric structure between Wasserstein and Bayes spaces, we believe that the choice between simplicial and Wasserstein frameworks is not trivial and should be application-driven.

To measure raw performance differences between geodesic and projected PCAs, we simulate data so that there is little recognizable structure in them, unlike in the previous example. The data generating process is as follows:

$$\begin{aligned} p_i(x) &\propto \sum_{j=1}^{K} w_{ij}\frac{1}{\sigma_{ij}}\exp\left((x-\mu_{ij})^2/(2\sigma_{ij}^2)\right)\mathbb{I}(x\in[-10,10]) + 10^{-5}, \quad i = 1,\dots 100, \\ \boldsymbol{w}_i &\sim \text{Dirichlet}_K(1/K), \\ (\mu_{ij},\sigma_{ij}) &\sim \mathcal{N}(d\mu_{ij};0,2^2)\text{Uniform}(d\sigma_{ij},0.5,2.0). \end{aligned} \tag{11.21}$$

Observe that (11.21) is a finite dimensional approximation of the Dirichlet Process mixture model, a popular workhorse in Bayesian nonparametric statistics, that is well known to be dense in the space of densities on $\mathbb{R}$, see for instance Ferguson (1983). An example of the kind of pdfs generated from (11.21) is shown in Figure 11.7.3(a).

To separate the effect of the B-spline smoothing procedure, in this scenario we evaluate the reconstruction error in (11.20) considering $\widetilde{\mu}_i$ to be the reconstructed quantile functions (for the Wasserstein PCAs) or pdfs (for the simplicial PCA) and μ_i to be the



(a)

(b)

Figure 11.7.3: Left panel: example of simulated data set for Scenario 2. Right panel: reconstruction error as a function of the dimension of the principal component employed for the different methods. The solid lines represent the mean of 10 independent runs on independent data sets from (11.21) and the shaded area represents ± one standard deviation.

probability measure represented by the B-spline approximation of the quantile function or the (centered log ratio of) the pdf respectively.

Figure 11.7.3(b) shows the reconstruction error as a function of the dimension of the principal component, that is, RE_k as a function of k. We can see how the three Wasserstein PCAs consistently outperform the simplicial one. Moreover, as to be expected, the global geodesic PCA obtains the lowest reconstruction error for all the choices of dimension k, with the nested geodesic PCA being a close runner-up. However, the computational cost of finding the nested or global geodesic PCA can become prohibitive as the sample size or the number of bases in the B-spline expansion or the dimension k increases. For comparison, finding the 10-dimensional projected PCA is around 1,000 times quicker than finding the corresponding global geodesic PCA and 200 times quicker than finding the nested geodesic one.

As an additional simulation, in Appendix 11.C we investigate the effect of the number of B-spline basis J. In particular, we conclude that, for a fixed dimension k the reconstruction error (11.20) increases with the number of basis functions, both for the projected and the simplicial PCA. Furthermore, we also observe that the reconstruction error for the simplicial PCA exhibits a larger variance than the reconstruction error for the projected PCA. Our insight is that this is due to the different degree of smoothness of the pdfs and the quantile functions. Since the quantile functions are in general smoother than the pdfs, their B-spline expansion should have lower variance.

11.7.2 Assessing the reliability of the projected PCA

A classical measure of performance of the standard Euclidean PCA, also useful to determine the dimension of the principal component to use, is the proportion of the explained variance. For a k-dimensional Euclidean principal component, this quantity is easily computed as a ratio of eigenvalues: $\sum_{j=1}^{k} \lambda_j \big/ \sum_{j \geq 1} \lambda_j$. Upon truncating the series at the denominator, the same quantity can also be computed for PCA in infinite dimensional Hilbert spaces.

Due to the projection step involved in our definition of PCA, we argue that the proportion of explained variance might not be a reliable indicator of performance, nor should it be used to guide the choice of the dimension k. Instead, we propose a fast alternative based on the Wasserstein distance that we believe better represents the properties of the projected PCA, that is, the normalized reconstruction error:

$$NRE_k = \frac{\frac{1}{n}\sum_{i=1}^{n} W_2(F_i^-, \widetilde{F}_i^-)}{\frac{1}{n}\sum_{i=1}^{n} W_2(F_i^-, F_0^-)},$$

where the numerator corresponds to the reconstruction error in (11.20) and the denominator is the average distance between the observed measures and their barycenter. Observe that in Euclidean spaces, this quantity is closely related to the proportion of explained variance, since, in Euclidean spaces, maximizing variance in a subspace amounts to minimizing the average distance from the subspace to data points.

Given its extrinsic nature, for a fixed dimension, the projected PCA might sometimes fail to capture the variability of some particular data set and, in those situations, an intrinsic approach should be preferred. However, given the high computational cost associated to geodesic PCAs, one would carry out such analysis only knowing that the results would be significantly better than the ones obtained by projected PCA. This calls for discerning whether the poor performance of projected PCA is due to its extrinsic nature or rather to the scarceness of structure in the data set under consideration: in the former situation it is likely that a geodesic approach would yield better results, in the latter instead, it is likely that results remain the same.

We propose now two empirical indicators of the 'reliability' of the empirical projected PCA. The first one measures, once a k-dimensional principal component is found, how reliable are the projected principal directions and the second one gives an idea of how different the projected PCA and the L_2 PCA are. To assess the interpretability of the principal directions and the scores obtained with the projected PCA, we first compute for every principal direction $\boldsymbol{w}_h^*$ the quantities $\eta_h^{\min}$ and $\eta_h^{\max}$ such that

$$\eta_h^{\min} = \min_{\eta \in \mathbb{R}} \{\boldsymbol{a}_0 + \eta \boldsymbol{w}_h^* \in \mathbb{R}^{J\uparrow}\},$$

where $\boldsymbol{a}_0$ is the spline coefficient vector associated with the barycenter F_0^-. The scalar $\eta_h^{\max}$ is found analogously. Hence $(\eta_h^{\min}\boldsymbol{w}_h^*, \eta_h^{\max}\boldsymbol{w}_h^*)$ is the segment spanned by the principal direction living inside the convex cone $\mathbb{R}^{J\uparrow}$. If the scores of all observations along this direction lie within the range $(\eta_h^{\min}, \eta_h^{\max})$, then the variability captured by (empirical) projected PCA can be decomposed along the principal directions, whose scores are then highly interpretable. Contrary, the PCA scores outside $(\eta_h^{\min}, \eta_h^{\max})$ will be associated with functions that are not quantiles, and thus limiting the interpretability of the direction. Hence, we propose the following *interpretability score*

$$IS_h = 1 - \frac{1}{n}\sum_{i=1}^{n} d\left(s_{ih}, [\eta_h^{\min}, \eta_h^{\max}]\right) / |s_{ih}|, \tag{11.22}$$

where s_{ih} is the score of observation i along direction h according to the projected PCA. A value of IS_h equal to one corresponds to perfect interpretability, that is, projected PCA behaves like a standard Euclidean PCA along direction h. On the other hand, values of IS_h closer to zero indicate that the decomposition of the variance along the principal directions lies outside $\mathbb{R}^{J\uparrow}$ for direction h. The interpretability score can be fruitfully used also to evaluate the directions found with the nested PCA, upon replacing the s_{ih}'s in (11.22) with the scores given by the nested PCA.

Note that the IS_h score is useful to interpret the directions one at a time. However, it can be the case that some scores along one direction h' lie outside the $(\eta_{h'}^{\min}, \eta_{h'}^{\max})$ range but that the L_2 projection on the $h \geq h'$ component still lies within the projected component. For instance, this could imply that a projected PC could be similar to a nested one despite having very different directions. A discrepancy between the two can appear when the projections of some data points on the L_2 PCA lie outside $\mathbb{R}^{J\uparrow}$. Using the terminology of Proposition 11.2 this can be measured in terms of difference between the projections $\Pi_k(F^{-*} - F_0^-)$ and $\Pi_{Sp(U_k)\cap(X-x_0)}(F^{-*} - F_0^-) = \Pi_{Sp(U_k)\cap(X-x_0)}(\Pi_k(F^{-*} - F_0^-))$, for a given observation F^{-*}. To quantify the loss of information at the level of the component (instead of direction), we propose to measure the 'ghost variance' captured by the L_2 PCA:

$$GV_k = \frac{1}{n}\sum_{i=1}^{n} \|\Pi_k(F_i^- - F_0^-) - \Pi_{U_X^{F_0^-},k}(\Pi_k(F_i^- - F_0^-))\|_2 \Big/ \|F_i^- - F_0^-\|_2,$$

that is, the GV_k score measures the quantity of information that is lost due to the projection step or, in other words, the information that we trained our PCA on, but that does not appear in the Wasserstein Space. If $GV_k = 0$ then all the information captured by the L_2 PCA is inside the Wasserstein Space, then the projected PCA coincides with the nested one by definition.

Finally, although this situation never occurred in our experience, it might happen that GV_k is small but some IS_k' $(k' \leq k)$ is large. This means that the subspace identified by the projected PCA is suitable for representing the data, but the single principal directions are not interpretable. In this case, we suggest taking a hybrid approach: use the projected PCA as a fast black-box dimensionality reduction step, thus reducing the dimensionality of each observation from J to k, and then use the nested PCA, in dimension k, to estimate the directions, the main advantage being the reduction in the computational cost to estimate the nested PCA in this lower dimensional space.

11.7.3 Analysis of the Covid-19 mortality data set

We perform PCA analysis on the Covid-19 mortality data publicly available at data.cdc.gov as of the first December 2020. The data set collects the total number of deaths due to Covid 19 in the US from January 1st, 2020 to the current date, data are subdivided by state, sex, and age. In particular, the ages of the deceased are grouped in eleven bins: $[0,1), [1,5), [5,15), [15,25), [25,35), [35,45), [45,55), [55,65), [75,85), [85,+\infty)$ but we truncate the last bin to 95 years for numerical convenience. Further, we remove Puerto Rico from the analysis because it presented too many missing values. Our final data set, shown in Figure 11.7.4(a), consists of 106 samples of the distribution of the ages of patients deceased due to Covid-19, divided by sex and pertaining 53 between US states and inhabited territories.

We apply our usual B-spline approximation with $J = 20$ basis to the quantile functions obtained starting from the histograms in Figure 11.7.4. This choice of J yields an average approximation error, in terms of Wasserstein distance, of 0.02. An error this low is to be expected since the quantile functions are piecewise linear functions defined on eleven intervals.

We use this real data set to make a hands-on comparison of the inference that can be obtained employing the projected, nested and log PCA.

We start by comparing the projected and nested PCAs. The first direction found by the nested PCA is identical to the one found by the projected while the second is extremely close: the cosine between the two principal directions is approximately 0.99. In line with this, the interpretability scores equal $IS_1 = 1$ and $IS_2 \approx 0.89$, while $GV_2 = 0.05$. Moreover, the two-dimensional projected principal component explains more than 90% of the L_2 variability and $NRE_2 \approx 0.05$ for both projected and nested PCA. Given the reconstruction error and the GV_2 score, we can conclude that the two-dimensional projected principal component provides a very good fit to the data, and that both selected principal directions are well behaved with respect to their scores, guaranteeing interpretable results.

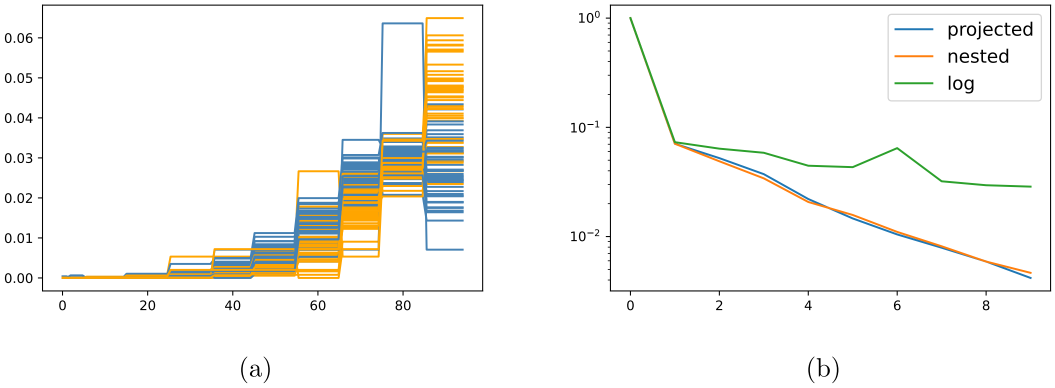


Figure 11.7.4: Left panel: distributions of age at the time of death for Covid-19 patients divided by sex: orange corresponds to females and blue to males. Different lines correspond to different US states / inhabited territories. Right panel: reconstruction error as a function of the dimension of the component for different PCAs. The 0-th principal component is the empirical mean.

Considering the discussion above and the fact that both the projected and nested PCA employ metric projection to map data points to the k-dimensional principal component, inference obtained with the nested PCA and with the projected one is almost identical in this case. We show results only for the projected PCA in Figure 11.7.5. In particular, the first principal direction shows that the greatest variability is due to the elders: low negative values along this direction correspond to most of the mortality being concentrated among in the 80+ range. The red and the green distributions displayed in the rightmost panel show two antithetic behaviors which correspond to scores along the first principal direction of roughly -8.5 and 7 as shown in the third panel of Figure 11.7.5. In fact, the red distribution is concentrated almost exclusively on the last two bins of the histogram, with the 85+ bin weighting for more of 60% of the deaths. On the opposite, the green distribution gives more weight to lower age values. The second direction instead shows variability in the $40-80$ range. The purple distribution, characterized by the highest score along this direction, shows that a significant percentage of deaths occurred in the age range $60 - 75$. Finally, the third panel of Figure 11.7.5 reports the scores along the first two principal directions for the whole data set, blue dots representing males and orange dots women. We can appreciate how women tend to have lower scores on both directions. This is in line with our understanding that Covid-19 is more severe among the male population (see for instance Mandavilli, 2020), which explains why males are more susceptible to death even at younger ages, while deaths among women are more concentrated in the 70+ age range, being the elders more fragile in general.

The comparison with log PCA requires more attention. First of all, note that the directions obtained with the projected and log PCA are the same by definition since they are both obtained performing PCA in $L_2([0,1])$, but the principal components may differ because different projection operators are employed when the orthogonal projection of a point onto the principal component lies outside of the image of φ_μ, as discussed in Section 11.3.4. As expected from the comparison between the metric projection and the pushforward operator in Figure 11.3.2, the fit to the data of the projected and log PCAs will be different. In particular, in this case, we observe that the log PCA does a worse job in terms of NRE, as shown in Figure 11.7.4(b), especially when the dimension increases. This behavior can be also partly explained by the complexity of the numerical routines needed to approximate the pushforward operator (required by the log PCA), where it is

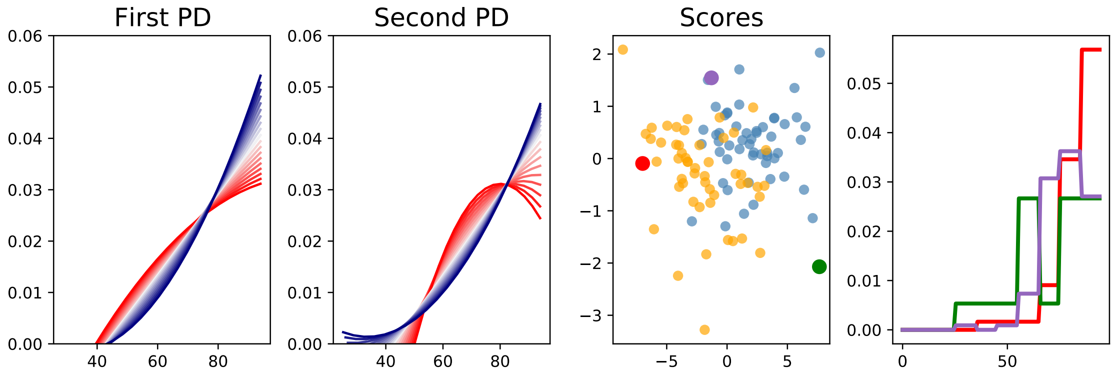


Figure 11.7.5: The first two panels show the variability along the first two principal directions (first and second panel), using the same visualization technique as in Figure 11.7.2. The third panel reports the scores of the projections on the two dimensional principal component (orange for women and blue for men) and the fourth panel shows three particular distributions, also highlighted in the third panel. In particular, the red distribution is the one of women in Vermont, the green one are males in Alaska and the purple one are women in West Virginia.

natural to expect some numerical errors.

More in general, as also discussed in Cazelles et al. (2018), we can conclude that the log PCA is not suited to study this particular data set because the L_2 PCA is different from the nested geodesic PCA (as testified by the GV_2 score). In fact, apart from the visual inspection of the L_2 principal directions – which are not guaranteed to span the log-principal components – not much can be obtained from the log PCA in this case since it does not provide a consistent way of projecting data points on the principal component as pointed out in Section 11.3.4.

11.8 Numerical Illustrations for the Distribution on Distribution Regression

In this section, we propose a comparison between the Wasserstein projected and simplicial (see Appendix 11.B) approaches when the task at hand is distribution on distribution regression and show an application of the Wasserstein projected regression framework to a problem of wind speed forecasting.

11.8.1 Simulation Study

We consider two data generating processes as follows. In the first setting, data are generating from the Wasserstein regression: independent variables $z_1, \dots, z_n$ are generated by considering quantile functions $F^-_{z1}, \dots, F^-_{zn}$ such that $F_{zi} = \sum_{h=1}^{30} a^{(z)}_{ih} \psi^{(3)}_j$ where $\psi^{(3)}_1, \dots, \psi^{(3)}_{30}$ is a cubic spline basis over equispaced knots in $[0, 1]$ and $a^{(z)}_{i1} = 0$, $a^{(z)}_{i2} = \delta_{i1}$, $a^{(z)}_{ij} = a^{(z)}_{ij-1} + \delta_{ij-1}$, and $(\delta_{i2}, \dots, \delta_{i30}) \sim \text{Dirichlet}(1, \dots, 1)$. This data generating procedure ensures the $F^-_{zi}(0) = 0$, $F^-_{zi}(1) = 1$ and F^-_{zi} is monotonically increasing, cf. Proposition 11.4. The dependent variables $F^-_{y1}, \dots, F^-_{yn}$ are generated using the same spline expansion of the dependent variables and letting $\boldsymbol{a}^{(y)}_i = B\boldsymbol{a}^{(z)}_i$. B is a randomly generated matrix with rows $\boldsymbol{b}_1, \dots, \boldsymbol{b}_{30}$, and each $\boldsymbol{b}_i$ is generated as follows: $b_{i1} \sim \text{Uniform}(0, 0.5)$ $b_{ij} = b_{ij-1} + \tilde{b}_{ij}$ and $\tilde{b}_{ij} \sim \text{Uniform}(0, 0.5)$, so that the coefficients $a^{(y)}_{ij}$ are monotonically

	First scenario	Second scenario
Wasserstein	$(4 \times 10^{-7}, 7 \times 10^{-8})$	$(5 \times 10^{-3}, 6 \times 10^{-3})$
Simplicial	$(0.9, 2.66)$	$(4 \times 10^{-4}, 5 \times 10^{-4})$

Table 11.8.1: Cross validation (leave one out) errors and standard deviations for the Wasserstein and Simplicial regression under the two simulated examples

non decreasing for each i and thus the F_{yi}^{-}'s can be considered quantile functions.

We compute the pushforward of the uniform distribution via numerical inversion and differentiation and obtain the pdf associated to each quantile function. Observe that this task is easier than approximating the pushforward of a generic μ through a generic f (as Cazelles et al. (2018) do) since the quantile functions are monotonic and we have simple expressions for all the quantities related to μ. Since the simplicial regression takes as input (a transformation of) the pdfs while the Wasserstein regression works directly on the quantile functions, and also due to the fact that numerical errors can be introduced in the data set during the inversion and differentiation, we consider as ground truth the pdfs and, for the Wasserstein approach, re-compute numerically the quantile functions.

In the second setting, instead, we generate data from the simplicial regression model: independent variables $z_1, \ldots, z_n$ are generated by applying the inverse of the centered log ratio to a random spline expansion as follows. For each $i = 1, \ldots, n$ let $\tilde{p}_{zi} = \sum_{j=1}^{30} a_{ij}^{(z)} \psi_j^{(3)}$ where the $\psi_j^{(3)}$'s are the same B-spline basis as in the previous setting. Here, the $a_{ij}^{(z)}$'s are generated iid from a Gaussian distribution with mean 0 and standard deviation 0.2. The dependent variables are generated by letting $\tilde{p}_{yi} = \sum_{j=1}^{30} a_{ij}^{(y)} \psi_j^{(3)}$ and $\boldsymbol{a}_i^{(y)} = B\boldsymbol{a}_i^{(z)}$, where B is a randomly generated 30×30 matrix with entries drawn iid from a standard normal distribution. Finally the pdfs p_{zi} (respectively p_{yi}) are recovered by applying the inverse of the centered log ratio to $\tilde{p}_{zi}$ (respectively $\tilde{p}_{yi}$), see Appendix 11.B for more details.

Note that under the second data generating process, both the dependent and independent distributions have support in $[0, 1]$ by construction, whereas, under the first data generating process, the independent variables might have a larger support. Thus, to fit the simplicial regression in the first scenario, as common practice (cf. Appendix 11.B), we extend the support of all the distributions (both dependent and independent) to the smallest interval of the real line containing all the supports. This is done by adding a small term to the pdfs (in our example, 10^{-12}) and then renormalizing them.

For both examples, we simulated 100 observations and compared the projected Wasserstein and simplicial regression using leave-one-out cross-validation. In particular, for both approaches, we use $J = 20$ quadratic spline basis and choose the penalty term ρ in (11.16) through grid search. Table 11.8.1 shows the pairs of mean squared error and standard deviation of the cross validation, the metric to compare the ground truth and the prediction is the 2-Wasserstein distance. As one might expect, the Wasserstein regression performs better in the first scenario, while the simplicial regression performs better in the second scenario. However, it is surprising how the Wasserstein geometry can capture (in terms of Wasserstein metric) dependence generated by a linear structure which we have shown to be very different from the Wasserstein one, making the projected regression a promising tool for such inferential problems

11.8.2 Wind speed distribution forecasting from a set of experts

We consider the problem of forecasting the distribution of the wind speed nearby a wind farm from a set of experts. The data set is publicly available at www.kaggle.com/theforcecoder/wind-power-forecasting. In particular, data consists of measurements of the wind speed collected every ten minutes for a period of 821 days starting from the 31st

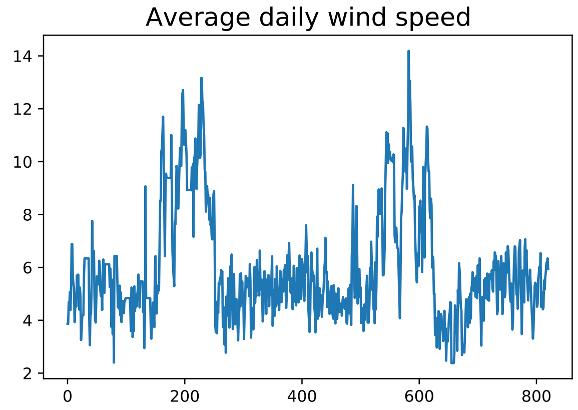


Figure 11.8.1: Daily average wind speed

December 2017. The daily average wind speed is shown in Figure 11.8.1.

We assume to have access to a set of *experts*, that is a set of trained models that provide a probabilistic one-day-ahead forecast for the average wind speed. Here, our goal is to combine this set of experts and provide a point estimate of the wind speed distribution for the whole day, which can be helpful when planning the maintenance of the wind mills, for instance.

Formally, let K denote the number of experts considered, F^{-}_{zij} is the quantile function associated with the probabilistic forecast of the average wind speed for day i given by expert $j = 1, \dots, K$; F^{-}_{yi} is the empirical quantile function of the wind speed for day i. In particular, we consider $K = 4$ experts built from the *Prophet* model by Facebook (Taylor and Letham, 2018) as follows: model $M1$ is the classical Prophet, without additional covariates or seasonality trends; model $M2$ includes the ambient temperature as covariate but not seasonality; model $M3$ includes a yearly seasonality and no covariates, and model $M4$ includes both yearly seasonality and ambient temperature as covariate. The models are estimated using variational inference on rolling samples of 365 days and produce one day ahead probabilistic forecasts for the average wind speed. The final sample size corresponds to $n = 456$.

We consider a trivial extension of the distribution on distribution regression model in Section 11.5.2 as follows:

$$\mathbb{E}[F^{-}_{yi} \mid F^{-}_{zi1}, \dots, F^{-}_{ziK}] = \Pi_{L_2([0,1])^{\uparrow}} \Big(\alpha + \sum_{j=1}^{K} \int_0^1 \beta_j(t, s) F^{-}_{zij}(t) \, \mathrm{d}t \Big). \tag{11.23}$$

Having approximated all the functions through a B-spline expansion, the model reads

$$\mathbb{E}[\boldsymbol{a}^{(y)}_i \mid \boldsymbol{a}^{(z)}_{i1}, \dots, \boldsymbol{a}^{(z)}_{iJ}] = \Pi_{\mathbb{R}^{J\uparrow}} \Big(\boldsymbol{\theta}_\alpha + \sum_{j=1}^{K} \Theta_{\beta_j} E \boldsymbol{a}^{(z)}_{ij} \Big).$$

The procedure for estimating $\boldsymbol{\theta}_\alpha$ and $\Theta_{\beta_1}, \dots \Theta_{\beta_K}$ is analogous to the one outlined in Section 11.5.2.

We compare the prediction performance of five distribution on distribution regression models. Models $R1$ to $R4$ are obtained by fitting model (11.23) using only one of the four experts, $M1$ to $M4$, while the fifth model (RF) is the 'full' model in (11.23) considering all the four experts. For this comparison, we perform a train-test split of the 456 days for which the experts produced the prediction, considering the last 100 days as test. We select

	$R1$	$R2$	$R3$	$R2$	RF
MSE	(1.22 ± 1.32)	(1.19 ± 1.26)	(1.15 ± 1.07)	(1.24 ± 1.23)	(0.86 ± 0.82)

Table 11.8.2: Mean square prediction error ± one standard deviation on the held-out test set.

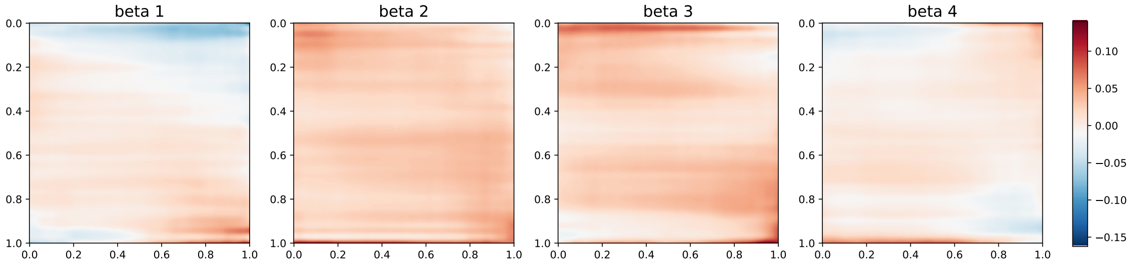


Figure 11.8.2: Estimates of the $\beta_i(t, s)$'s evaluated on $[0, 1]^2$. The variable t runs across columns, and variable s across rows

hyperparameters (namely, the penalty coefficient ρ in (11.16) and whether to include or not the intercept term α) by a grid search cross validation on the training set, and compare the mean square error on the held-out test set. Results of the comparison are reported in Table 11.8.2. As expected, the model with the four predictors (RF) is the best performer. Interestingly, all the other models $R1$-$R4$ perform similarly and present a much higher mean square error when compared to RF, thus suggesting that the best performance is achieved by combining the different experts together and no expert alone can be a good predictor. This is possibly explained by some experts being able to better forecast one scenario (for instance, light winds) and other experts being able to better forecast other scenarios.

We conclude with some descriptive analysis. Figure 11.8.2 shows the point estimates for the coefficients β_j. We can interpret as highly influential for the regression the areas of the β_j's with high absolute value and as negligible areas with values close to zero.

We can highlight some differences among the coefficients in Figure 11.8.2. In particular, model $M1$, seems influent when predicting the tails of the distribution, in particular with negative weights for the left tail and positive weights for the right tail. Model $M2$ seems to be affecting all the steps of the prediction and, in particular, to be the model affecting the most the median of the distribution. Model $M3$ appears to be, with $M2$, the most important model for the prediction: the absolute value in the corresponding regressor β_3 is often very high and with noticeable peaks corresponding to areas predicting the left tail and towards the right tail. Finally, the regressor corresponding to $M4$ has very low values, thus resulting in minor importance in terms of regression influence.

Interestingly, the experts providing the most precious inputs to our regression model are $M2$ and $M3$, that incorporate only the seasonality effect and the temperature covariate respectively, while $M4$, which incorporates both, seems to be less important. Hence, the regression model in (11.23) finds more effective combining experts trained on different covariates than correcting an expert already trained on all the covariates. In particular, our insight is that $M2$ is responsible for centering the median of the output distribution. The tails of the distribution seem to need also the contribution of seasonality data, given by $M3$. Finally, we also observe that the left tail of the wind distribution seems the most difficult to be predicted, needing very high positive and negative weights across different models to be obtained.

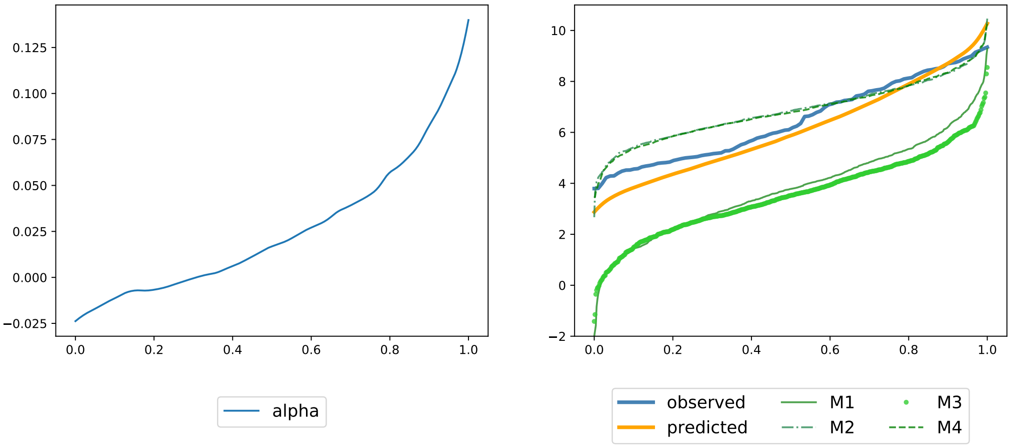


Figure 11.8.3: Estimate of α (left) and prediction of one F_y^- of the test set (right). In the right panel, the blue line corresponds to the empirical quantile function, the orange one to the prediction from RF and the green ones to the average wind predictions obtained from the experts $M1$-$M4$.

11.9 Discussion

In this chapter, we propose a novel class of *projected* statistical methods for distributional data on the real line, focusing in particular on the definition of a *projected* PCA and a *projected* linear regression. By investigating the weak Riemannian structure of the Wasserstein space and the transport maps between probability measures, we represent the Wasserstein space as a closed convex cone inside an Hilbert space.

Similar to *log* methods, our models exploit the possibility to map data into a linear space to perform statistics in an *extrinsic* fashion. However, instead of using operators like the *exp* map or some kind of boundary projection to return to the Wasserstein space, we rely on a metric projection operator that is more respectful of the underlying metric.

By choosing as base point the uniform measure on $[0, 1]$, we are able to efficiently approximate the metric projection operator so that our models combine the ease of implementation of *extrinsic* methods while retaining a performance similar to the one of *intrinsic* methods. Further, through a quadratic B-spline approximation, we can greatly reduce the dimensionality of the optimization problems involved, resulting in fast empirical methods. As a byproduct of this approach, we also derive faster numerical routines for the *geodesic* PCA in Bigot et al. (2017).

We study asymptotic properties of the proposed methods, concluding that, under reasonable regularity assumptions, our *projected* models provide consistent estimates and that the B-spline approximation error becomes negligible. We showcase our approach in several simulation studies and using two real world data sets, comparing our models to *intrinsic* and *extrinsic* ones and to the *simplicial* approach in Hron et al. (2014), concluding that the *projected* PCA and regression constitute a valid candidate for performing inference on a data set of distributions.

Although our *projected* framework was proven to be viable in many practical situations, some care must be taken when adopting it, especially when performing PCA. In fact, the *extrinsic* nature of our method might not fit every data set, in which case a more computationally demanding *intrinsic* PCA might be preferred, see for instance Appendix 11.D.1 for an example where the *projected* principal directions are not interpretable. On top of that, performing PCA in the Wasserstein space requires more attention than performing the usual Euclidean PCA: as pointed out in Appendix 11.D.2, since principal components are not linear subspaces, decomposing the variance along the directions (i.e., looking at the scores) must be done carefully, and making sure that the directions are indeed interpretable. To assist practitioners, in Section 11.7.2 we have also proposed two scores that quantify the interpretability of the principal directions and the discrepancy between the *nested* and *projected* principal components.

Several extensions and modifications of our approach are possible. One possibility is to extend our framework to encompass more models, such as generalized linear models and independent component analysis. Although this should be straightforward in theory, the numerical computations could become more burdensome. Furthermore, as an alternative to our approach based on B-splines approximation, one could use such B-spline expansion only to approximate the metric projection operator. Another interesting line of research would consist in building hybrid approaches (as anticipated in Section 11.7.2) to analyze distributions in the Wasserstein space, using both *extrinsic* and *intrinsic* methods to exploit the advantages of both worlds while mitigating the disadvantages. We also think that a deeper comparison between the Wasserstein and the simplicial geometries could help practitioners in choosing between them.

Finally, as pointed out by an anonymous referee, extensions to encompass measures supported on $\mathbb{R}^d$, $d > 1$, are of great interest. This is surely a very challenging problem due to the geometric structure of $\mathcal{W}_2(\mathbb{R}^d)$. We identify three main obstacles in this sense. First, the map onto the tangent space is not an isometry because the Wasserstein space is curved. Second, we lose the nice characterization of the tangent space and of the image of $\log_\mu$, so that the metric projection operator becomes harder to derive. Third, the computational cost would greatly increase due to the need of approximating numerically the transport maps needed to compute the distances.

Appendix

11.A Proofs

<u>*Assumptions on x_0.*</u>

Let $B_\varepsilon(x_0) = \{x \in H \,\big|\, ||x - x_0|| < \varepsilon\}$, a ball of radius ε in H. Given a set C, we refer to aff(C) as the smallest affine subset containing C, found as the intersection of all affine subspaces containing C. Similarly $\mathcal{H}(C)$ is the convex hull of C, the smallest convex subset of H containing it. The relative interior of a set C is defined as its interior considering as ambient space aff(C): relint$(C) = \{x \in C \,\big|\, \exists B_\varepsilon(x_0)$ such that $B_\varepsilon(x_0) \cap \text{aff}(C) \subset C\}$.

Throughout our chapter we assume that the random variable $\mathcal{X}$ is such that (i) there exists $x_0 = \mathbb{E}[\mathcal{X}]$ and (ii) $x_0 \in \text{relint}(\mathcal{H}(\text{supp}(\mathcal{X})))$ where supp$(\mathcal{X})$ is the support of $\mathcal{X}$. These assumptions are indeed quite natural and require that the distribution of $\mathcal{X}$ has a well defined barycenter, which is not in a 'degenerate' position with respect to the convex hull of its support, which may happen in infinite dimensional Hilbert Spaces. See, for instance, Berezin and Miftakhov (2019) for an example of distributions not verifying this second assumption.

<u>*Proof of* Lemma 11.1.</u>

The proof is divided in two steps. First, we prove that $(x_0 + Sp(U_k)) \cap X$ has dimension k. Then, we show that $U_X^{x_0,k} = (x_0 + Sp(U_k)) \cap X$. Without loss of generality, for ease of notation, we perform an affine change of variable so that $x_0 = 0$, but, with a slight abuse of notation, we keep denoting with $\mathcal{X}$ and X the transformed random variable and the convex cone respectively.

To prove the first part, let $\mathcal{H}(\mathcal{X})$ be the convex hull of the support of $\mathcal{X}$ and aff$(\mathcal{H}(\mathcal{X})) = K$ be the smallest affine subset of H containing $\mathcal{H}(\mathcal{X})$. We know by assumption that there is an open ball in K which contains $x_0 = 0$ and is contained in $\mathcal{H}(\mathcal{X})$. Moreover, for every $k \leq dim(K)$, $Sp(U_k) \subset K$. Note that we can clearly suppose $k \leq dim(K)$, otherwise principal components analysis is useless. With this assumption, since $x_0 = 0$ is in the relative intern of $\mathcal{H}(\mathcal{X})$, we have $k = dim(Sp(U_k) \cap \mathcal{H}(\mathcal{X})) \leq dim(Sp(U_k) \cap X) \leq k$.

Now we prove that a $(k, 0)$-projected principal component is given by $Sp(U_k) \cap X$. To prove this, let C^* be a $(k, 0)$-projected principal component and $A^* = A \cap X$, with $A = Sp(U_k)$: we know (i) $x_0 = 0 \in A^*$, (ii) $dim(A^*) = k$ by definition and (iii) $A^* \subseteq \Pi_X(A)$, so we have $A^* \subset C^*$.

Since $dim(C^*) = k$ there is C linear subspace of dimension k such that $C^* \subset C$. Consider $C' = C \cap X$: clearly $C^* \subset C'$, so that $A^* \subset C^* \subset C'$. Moreover, $A^* \subset C'$, which implies $A \cap X \subset C \cap X$ and thus $Sp(A \cap X) \subset Sp(C \cap X)$. The proof is concluded if $dim(Sp(A \cap X)) = dim(Sp(C \cap X)) = k$. In fact, in this case $A = Sp(A \cap X)$ and $C = Sp(C \cap X)$ which means that $A \subset C$ and since $dim(A) = dim(C) = k$, A and C coincide, proving $A^* = C^*$.

To prove this final claim, observe that $dim(Sp(A \cap X)) < k$ implies $dim(A \cap X) < k$, which contradicts the proof of the first part of this Lemma. Similarly, $dim(Sp(C \cap X)) = k$ since $dim(C^*) = k$ by hypothesis.

■

Proof of Proposition 11.1.

The fact that $\|\Pi_{U_X^{x_0,k}}(x)-x\| \geq \|\Pi_{U_X^{x_0,k+1}}(x)-x\|$ follows easily by noticing that $U_X^{x_0,k} \subset U_X^{x_0,k+1}$.

To prove that $\|\Pi_{U_X^{x_0,k}}(x)-x\| \to 0$ as k increases, we notice that, by the properties of the principal components in H, we have $\Pi_{Sp(U_k)}(x-x_0) \xrightarrow{k} x-x_0$ for every $x \in X$, which implies $\|\Pi_{Sp(U_k)+x_0}(x)-x\| \to 0$. Denote $x_1 = \Pi_{U_X^{x_0,1}}(x)$ and let r_k be the line between x_1 and x. Let:

$$x_k = \underset{x' \in r_k \cap Sp(U_k)+x_0}{\arg\min} \|x'-x\|.$$

We clearly have have $x_k \to x$. Finally, by convexity we know $x_k \in U_X^{x_0,k}$, which implies $\|\Pi_{U_X^{x_0,k}}(x)-x\| \leq \|x_k - x\| \to 0$.

■

Proof of Proposition 11.2.

Without loss of generality, for ease of notation, we perform an affine change of variable so that $x_0 = 0$, but, with a slight abuse of notation, we keep denoting with $\mathcal{X}$ and X the transformed random variable and convex cone respectively.

We note that being Π_k the orthogonal projection onto a subspace, $x - \Pi_k(x) \perp Span(U_k)$ and thus for $v \in Span(U_k)$:

$$\|x^* - v\|^2 = \|x^* - \Pi_k(x^*)\|^2 + \|\Pi_k(x^*) - v\|^2.$$

Then

$$\underset{v \in U_X^{0,k}}{\arg\min} \|x^* - v\| = \underset{v \in Sp(U_k) \cap X}{\arg\min} \|\Pi_k(x^*) - v\|$$

and the result follows. ■

Proof of Proposition 11.4.

1. As shown in the supplementary of Pya and Wood (2015) by standard B-spline formulas we obtain that given $f(x) = \sum_{j=1}^{J} a_j \psi_j^k(x)$, then $f'(x) = \sum_{j=1}^{J}(a_j - a_{j-1}) \cdot \psi_j^{k-1}(x)$. Being the B-spline basis function nonnegative by definition, we obtain the result.

2. With $k=2$, $f'(x)$ on the interval $[x_{j+1}, x_j]$ has the following expression:

   $$\frac{x - x_j}{x_{j+1} - x_j} \cdot (\alpha_j - \alpha_{j-1}) + \frac{x_{j+1} - x}{x_{j+1} - x_j} \cdot (\alpha_{j-1} - \alpha_{j-2}),$$

   so:

   $$lim_{x \to x_{j+1}^-} f'(x) = \alpha_j - \alpha_{j-1}$$

   and the result follows. ■

Proof of Proposition 11.5 and 11.6.

We report here Propositions 3.3 and 3.4 of Bigot et al. (2017), with the notation adapted to our manuscript. In the following, H is a separable Hilbert space, X is a closed convex subset of H, $\mathcal{X}$ is an X-valued square-integrable random variable, x_0 a point in X and $k \geq 1$ an integer.

Proposition 11.10. *Let $U^* = \{u_1^*, .., u_k^*\}$ be a minimizer over orthonormal sets U of H of cardinality k, of $D_X^{x_0}(\mathcal{X}, U) := \mathbb{E}d^2(\mathcal{X}, (x_0 + Sp(U)) \cap X)$, then $U_X^{x_0} := (x_0 + Sp(U)) \cap X$ is a $(k, x_0)-$global principal component of $\mathcal{X}$.*

Proposition 11.11. *Let $U^* = \{u_1^*, .., u_k^*\}$ be an orthonormal set such that $U_i^* = \{u_1^*, .., u_i^*\}$ is a minimimizer of $D_X^{x_0}(\mathcal{X}, U)$ over the orthonormal sets of cardinality 'i' such that $U \supset U_{i-1}^*$; then $U_X^{*x_0}$ is a $(k, x_0)-$nested principal convex component of $\mathcal{X}$.*

Applying Propositions 11.10 and 11.11 we can obtain equivalent definitions of geodesic and nested PCA as optimization problems in $L_2([0,1])$. If we fix $J \in \mathbb{N} > 0$ and a quadratic B-spline basis $\{\psi_j\}_{j=1}^J$, we can use Propositions 11.10 and 11.11 with $X = L_2([0,1])^{J\uparrow}$ and $H = L_2([0,1])^J$. Thanks to Remark 11.7 we obtain the results.

■

Proof of Proposition 11.7.

Let $S_J = \sum_{j=1}^J \lambda_j^{(J)} \psi_j^{(J)}$ and its derivative $s_J = \sum_j (\lambda_j^{(J)} - \lambda_{j-1}^{(J)}) \widetilde{\psi}_j^{(J)}$ where $\widetilde{\psi}_j^{(J)}$ denotes the linear spline basis on the same equispaced grid in $[0,1]$.

Let $f_\mu^- = (F_\mu^-)'$. Of course, it can be seen that f_μ^- is non-negative. Moreover, it is obvious that $f_\mu^- \in W_2^\infty([0,1])$. Then, from De Boor and Daniel (1974) we get that there exist s_J such that $\|s_J - f_\mu^-\|_\infty \leq C\|D^2 f_\mu^-\|_\infty J^{-2}$, where C is a constant depending on the interval $[0,1]$ but not on n.

Hence, we can determine the coefficients $\{\lambda_j^{(J)}\}$, starting from the spline s_J, up to a translation factor.

We fix a particular set of coefficients by letting $S_J(0) = \lambda_1^{(J)} = F_\mu^-(0)$ for each J. So that:

$$S_J(x) - F_\mu^-(x) = \int_0^x s_J(t)dt - \int_0^x f_\mu^-(t)dt - S_J(0) + F_\mu^-(0) = \int_0^x s_J(t) - f_\mu^-(t)dt.$$

By using the previous result, the integral we have that $S_J(x) - F_\mu^-(x) \leq CJ^{-2}$ for all x which proves the proposition. ■

Proof of Proposition 11.8.

By the Assumptions in Section 11.6.2.1 and Remark 11.10 there exists a ball B_K in $W_3^\infty([0,1])$ of radius K for some $K > 0$, such that each F_i^- can be ε-approximated by $\widetilde{F}_i^- \in W_3^\infty([0,1])$ with $\widetilde{F}^- \in B_K$. We can suppose that also the eigenvectors of the covariance operator of the generating process belong to such sphere, otherwise we just increase its radius of some finite amount.

By Proposition 11.7 we can choose a spline basis (that is, a number of elements $J > 0$), such that we get a ε-uniformly good approximation of B_K (and thus we can 2ε-approximate its L_2 closure). To lighten notation, thanks to Remark 11.7 we deliberately confuse $\mathbb{R}^{J\uparrow}$ and the space monotone B-splines with J basis functions, the inner product we are referring to will always be clear by looking at its entries.

Now consider the following inequalities, with $\boldsymbol{a}_i^J$ obtained as 2ε approximations of F_i^-, $\boldsymbol{w}^J \in \mathbb{R}^{\mathbb{J}}$, $w \in L_2([0,1])$:

$$\Big|\frac{1}{n}\sum_i \langle F_i^-, w\rangle^2 - \frac{1}{n}\sum_i \langle \boldsymbol{a}_i^J, \boldsymbol{w}^J\rangle^2\Big| \leq$$
$$\frac{1}{n}\Big|\sum_i \langle F_i^-, w\rangle^2 - \sum_i \langle \boldsymbol{a}_i^J, w\rangle^2 + \sum_i \langle \boldsymbol{a}_i^J, w\rangle^2 - \sum_i \langle \boldsymbol{a}_i^J, \boldsymbol{w}\rangle^2\Big|,$$

where the inner product $\langle \boldsymbol{a}_i^J, w\rangle$ is to be intended as the L_2 inner product between the spline function with coefficients $\boldsymbol{a}_i^J$ and the L_2 function w. Consider now:

$$\begin{aligned}
&\frac{1}{n}\sum_i(\langle F_i^-, w\rangle^2 - \langle \boldsymbol{a}_i^J, w\rangle^2) = \\
&\qquad\qquad \frac{1}{n}\sum_i(\langle F_i^-, w\rangle - \langle \boldsymbol{a}_i^J, w\rangle)(\langle F_i^-, w\rangle + \langle \boldsymbol{a}_i^J, w\rangle) = \\
&\qquad\qquad \frac{1}{n}\sum_i\langle F_i^- - \boldsymbol{a}_i^J, w\rangle\langle F_i^- + \boldsymbol{a}_i^J, w\rangle \le \\
&\qquad\qquad \frac{1}{n}\sum_i \Big|\langle F_i^- - \boldsymbol{a}_i^J, w\rangle\Big| \cdot \Big|\langle F_i^- + \boldsymbol{a}_i^J, w\rangle\Big| \le \\
&\qquad\qquad \frac{1}{n}\sum_i 2\varepsilon\|w\|^2 2K = 4\varepsilon K\|w\|^2.
\end{aligned}$$

Similarly:

$$\Big|\frac{1}{n}\sum_i(\langle \boldsymbol{a}_i^J, w\rangle^2 - \langle \boldsymbol{a}_i^J, \boldsymbol{w}^J\rangle^2)\Big| \le \|\boldsymbol{a}_i^J\|^2 \cdot \|w - \boldsymbol{w}^J\| \cdot (\|w\| + \|\boldsymbol{w}^J\|).xx$$

We know that a solution to the problem $\max_{\|w\|_{L_2}=1}\frac{1}{n}\sum_i\langle F_i^-, w\rangle^2$ is given by the first eigenfunction $\widehat{w}$ of the covariance operator of the empirical process. Now we are in the condition to apply results in Dauxois et al. (1982), or in Qi and Zhao (2011) (with $\alpha \to 0$) to conclude that $\widehat{w}$ converges to the first eigenfunction $\bar{w}$ of the covariance operator of the process that generates F_i^-. By hypothesis, such eigenfunction $\bar{w}$ lies in B_K and thus can be approximated with our fixed spline basis. Thus for high enough n, also $\widehat{w}$ can be approximated up to 2ε.

Let $\boldsymbol{a}_{\widehat{w}}$ be the coefficients of the spline expansion of $\widehat{w}$ spline approximation, that is, $\|w - \boldsymbol{a}_w\| \le 2\varepsilon$. Observe that $\Big|\|\widehat{w}\|_2 - \|\boldsymbol{a}_{\widehat{w}}\|_E\Big| \le 2\varepsilon$, just as $\|\boldsymbol{a}_J^i\| \le K + 2\varepsilon$. Thus, up to adding another ε to the approximation error $\|\widehat{w} - \boldsymbol{a}_{\widehat{w}}\|$, we can suppose $\|\boldsymbol{a}_{\widehat{w}}\|_2 = 1$. Hence:

$$\Big|\frac{1}{n}\sum_i(\langle \boldsymbol{a}_i^J, \widehat{w}\rangle^2 - \langle \boldsymbol{a}_i^J, \boldsymbol{a}_{\widehat{w}}\rangle^2)\Big| \le (K + 2\varepsilon) \cdot 3\varepsilon \cdot 2,$$

which leads to:

$$\Big|\max_{\|w\|_{L_2}=1}\sum_i\langle \boldsymbol{a}_i^J, w\rangle^2 - \max_{\|\boldsymbol{w}^J\|_E=1}\sum_i\langle \boldsymbol{a}_i^J, \boldsymbol{w}^J\rangle^2\Big| \le (K + 2\varepsilon) \cdot 3\varepsilon \cdot 2.$$

Finally, combining the above results and the fact that $|\max f - \max g| \le \max|f - g|$ for any pair of real valued functions f and g, we obtain:

$$\begin{aligned}
&\Big|\max_{\|w\|_{L_2}=1}\frac{1}{n}\sum_i\langle f_i, w\rangle^2 - \max_{\|\boldsymbol{w}^J\|_E=1}\frac{1}{n}\sum_i\langle \boldsymbol{a}_i^J, \boldsymbol{w}^J\rangle^2\Big| \le \\
&\qquad\qquad\qquad\qquad \max_{\|w\|_{L_2}=1} 4\varepsilon K\|w\| + (K + 2\varepsilon) \cdot 6\varepsilon \le 6\varepsilon K(1 + 2\varepsilon).
\end{aligned}$$

Thus for instance if we ask that $\varepsilon < 1$, we obtain the desired result with $D = 18 \cdot K$. Consistency follows since $\|\boldsymbol{a}_{\widehat{w}} - \bar{w}\| \le \|\boldsymbol{a}_{\widehat{w}} - \widehat{w}\| + \|\widehat{w} - \bar{w}\|$. ■

Proof of Lemma 11.2.
Since for any $x \in X$ we have $\Pi_{\mathbb{R}^{J\uparrow}}(x) \to x$, for any $v \in H$:

$$\|v - \Pi_{\mathbb{R}^{J\uparrow}}(v)\| \leq \|v - \Pi_{\mathbb{R}^{J\uparrow}}(\Pi_X(v))\| \leq \|v - \Pi_X(v)\| + \|\Pi_X(v) - \Pi_{\mathbb{R}^{J\uparrow}}(\Pi_X(v))\|$$

which implies $\Pi_{\mathbb{R}^{J\uparrow}}(v) \to \Pi_X(v)$. Consider now $b_n \to b$ in H; we have the inequality:

$$\|\Pi_{\mathbb{R}^{J\uparrow}}(b_n) - \Pi(b)\| \leq \|\Pi_{\mathbb{R}^{J\uparrow}}(b_n) - \Pi_X(b_n)\| + \|\Pi_X(b_n) - \Pi_X(b)\|$$

the first term of the right-hand side of the inequality can be sent to 0 by increasing J, the other by increasing n. ■

Proof of Proposition 11.9.
We call a_i the spline coefficients associated to x_i and b_i the ones associated to y_i. Again we deliberately confuse the spaces where the coefficients and the spline functions live to lighten the notation. Since the penalty term does not depend on the data, we have:

$$\begin{aligned} &\frac{1}{n}\Big|\sum_i \|y_i - \langle x_i, B^T AB\rangle\|^2 - \sum_i \|b_i - \langle a_i, B^T AB\rangle_{L_2([0,1])}\|^2| = \\ &\quad\frac{1}{n}|\sum_i (\|y_i - \langle x_i, B^T AB\rangle\|^2 - \|b_i - \langle a_i, B^T AB\rangle_{L_2([0,1])}\|^2)| \leq \\ &\quad\frac{1}{n}\sum_i \|y_i - \langle x_i, B^T AB\rangle\|^2 - \|b_i - \langle a_i, B^T AB\rangle_{L_2([0,1])}\|^2|. \end{aligned}$$

Now, since

$$\begin{aligned} &\Big|\|y_i - \langle x_i, B^T AB\rangle\|^2 - \|b_i - \langle a_i, B^T AB\rangle_{L_2([0,1])}\|^2\Big| = \\ &\quad\Big|(\|y_i - \langle x_i, B^T AB\rangle\| - \|b_i - \langle a_i, B^T AB\rangle\|)\times \\ &\qquad(\|y_i - \langle x_i, B^T AB\rangle\| + \|b_i - \langle a_i, B^T AB\rangle\|)\Big|. \end{aligned}$$

Then for some constant K depending on the bounds in the Assumptions, we get:

$$\begin{aligned} &\Big|\|y_i - \langle x_i, B^T AB\rangle\|^2 - \|b_i - \langle a_i, B^T AB\rangle_{L_2([0,1])}\|^2\Big| \leq \\ &\quad\|y_i - \langle x_i, B^T AB\rangle - b_i + \langle a_i, B^T AB\rangle\|2K = \\ &\quad\big(\|y_i - b_i\| + \langle a_i - x_i, B^T AB\rangle\big)\, 2K. \end{aligned}$$

Thus, if J is such that we have ε-approximations of the data, by Cauchy-Schwartz we obtain:

$$\frac{1}{n}\Big|\sum_i \|y_i - \langle x_i, B^T AB\rangle\|^2 - \sum_i \|b_i - \langle a_i, B^T AB\rangle_{L_2([0,1])}\|^2\Big| \leq K' \cdot \varepsilon,$$

for some K' constant.

Thanks to the results in Prchal and Sarda (2007), for any $\varepsilon > 0$, if the number of samples is big, $\widehat{\Theta}$ and $\widehat{\Theta}_J$ exist with probability $1 - \varepsilon$ and are unique. Since the value of the minimization problem the solve are arbitrarily close, then the minimizers converge in $\mathbb{R}^{J\times J}$ with the metric given by the spline basis. ■

Strong convergence implies semi-norm convergence.

Let $\mathcal{Z}$ be an H-valued random variable and $\mathcal{C}_{\mathcal{Z}}$ the covariance operator associated to $\mathcal{Z}$, that is:

$$(\mathcal{C}_{\mathcal{Z}} f)(s) = \int_{[0,1]} cov(\boldsymbol{x}(s), \boldsymbol{x}(t)) f(t) dt.$$

In the following, we denote with $\| \cdot \|_{L_2}$ the $L_2([0,1]^2)$ norm. Further, recall that $\|cov(\mathcal{Z}(s), \mathcal{Z}(t))\|_{L_2([0,1]^2)} = \mathbb{E}[\|\mathcal{Z}\|^2]$. We want to look at the behavior of $\|\widehat{\beta}_{\mathrm{PS}} - \widehat{\beta}_J\|_{\mathcal{C}_{\mathcal{Z}}}$.

$$\begin{aligned}
&\int_{[0,1]} \langle \mathcal{C}_{\mathcal{Z}}(\widehat{\beta}_{\mathrm{PS}}(s,t) - \widehat{\beta}_J(s,t)), \widehat{\beta}_{\mathrm{PS}}(s,t) - \widehat{\beta}_J(s,t) \rangle dt \leq \\
&\qquad \|\mathcal{C}_{\mathcal{Z}}(\widehat{\beta}_{\mathrm{PS}}(s,t) - \widehat{\beta}_J(s,t))\|_{L_2} \cdot \|\widehat{\beta}_{\mathrm{PS}}(s,t) - \widehat{\beta}_J(s,t)\|_{L_2} \leq \\
&\qquad \mathbb{E}[\|\boldsymbol{x}\|^2] \cdot \|\widehat{\beta}_{\mathrm{PS}}(s,t) - \widehat{\beta}_J(s,t)\|_{L_2} \cdot \|\widehat{\beta}_{\mathrm{PS}}(s,t) - \widehat{\beta}_J(s,t)\|_{L_2}.
\end{aligned}$$

So $\|\widehat{\beta}_{\mathrm{PS}} - \widehat{\beta}_J\|_{\mathcal{C}_{\mathcal{Z}}} \leq M \cdot \|\widehat{\beta}_{\mathrm{PS}} - \widehat{\beta}_J\|^2_{L_2}$ for some constant M. Thus $\| \cdot \|_{L_2}$ convergence implies $\| \cdot \|_{\mathcal{C}_{\mathcal{Z}}}$ convergence.

11.B The simplicial approach

The simplicial approach to distributional data analysis is based on the definition of Bayes space $\mathcal{B}^2(I)$ (Egozcue et al., 2006). Formally, let $I \subset \mathbb{R}$ a closed interval, the Bayes spaces $\mathcal{B}^2(I)$ is defined the equivalence class of probability densities $p(x)$ on I (that is $p(x) \geq 0$ and $\int_I p(x)dx = 1$) with square integrable logarithm.

The Bayes space is endowed with a linear space starting from the definition of the perturbation and powering operators, that are analogous to the sum and multiplication times a scalar, and inner product. Moreover Menafoglio et al. (2014) defines an isometric isomorphism between $\mathcal{B}^2(I)$ and $L_2([0,1])$ through the so-called centered log ratio (clr) map defined as

$$\widetilde{p}(x) := \mathrm{clr}(p)(x) = \log(p(x)) - \frac{1}{b-a} \int_a^b \log p(t) dt, \tag{11.24}$$

for every $p \in \mathcal{B}^2(I)$. The inverse map is defined as

$$p(x) = \mathrm{clr}^{-1}(\widetilde{p})(x) = \frac{\exp(\widetilde{p}(x))}{\int_I \exp(\widetilde{p}(x))dx}.$$

Thus, it is possible to define a *simplicial* PCA and *simplicial* regression on the Bayes space starting from the clr map. In particular, let $p_1, \ldots, p_n$ be observed densities on the interval I and let $\widetilde{p}_i = \mathrm{clr}(p_i)$. Denote with $\widetilde{w}_1, \ldots, \widetilde{w}_k$ the first k principal directions estimated from the $\widetilde{p}_i$'s, then a k dimensional simplicial principal component is the span of $\{w_i = \mathrm{clr}^{-1}(\widetilde{w}_i)\}_{i=1}^k$ in $\mathcal{B}^2(I)$.

Similarly, for pdfs $\{(p_z, p_y)_i\}_{i=1}^n$ a simplicial regression model is defined starting from the clr transformed variables. Let $\widetilde{\Gamma}$ denote a functional regression model in L_2 for variables $\{(\widetilde{p}_z, \widetilde{p}_y)_i\}_{i=1}^n$, then the simplicial regression states:

$$\mathbb{E}[p_{yi} \,|\, p_{zi}] = \mathrm{clr}^{-1}\left(\widetilde{\Gamma}(\widetilde{p}_{zi})\right).$$

Apart from the different geometries of the Wasserstein and Bayes space, which are discussed in Sections 11.7 and 11.8, we can highlight one particular drawback from the simplicial approach, which we believe poses a significant limit to its usefulness. In fact, the main assumption is that all the pdfs p_i share the same support, which might not be the case (for instance, it is not the case for our example in Section 11.8.2). In practice,

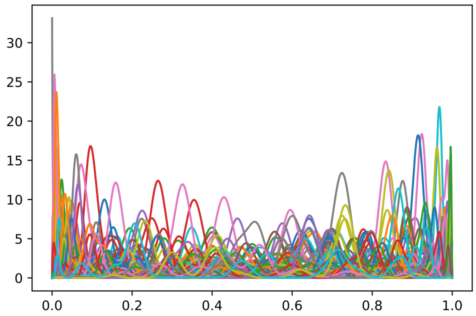


Figure 11.C.1: Example of data set from (11.26)

one may circumvent this need by either 'padding' all the pdfs to the same support, i.e considering

$$\overline{p}_i(x) \propto p_i(x) + \varepsilon \mathbb{I}[x \in I], \tag{11.25}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function, and the proportionality is due to the need of re-normalizing the $\overline{p}_i$'s so that they integrate to 1. Another approach could consist in considering I as the intersection of all the supports of the different p_i's let truncate all the pdfs to the shared interval I.

Both approaches present undesired side effects that can greatly alter the results. The second approach might end up with a very small interval I, so that a lot of information is lost due to this pre-processing step. The drawback of the first approach instead is due to numerical instability. In fact, one would like ε in (11.25) to be small in order not to corrupt the true signal, given by p_i. However, considering the transformation in (11.24) having a small ε would cause the $\widetilde{p}_i$ to present some extreme values (negative) in correspondence to ε. Performing PCA on a data set processed in this way would greatly alter the results, as most of the variability of the $\widetilde{p}_i$'s would be masked by a difference in their support.

11.C Additional Simulations

11.C.1 Sensitivity Analysis to the Number of Basis Functions

In this simulation, we show how the number of B-spline basis functions affects the inference in our projected PCA and in the simplicial one. In this scenario, the probability measures are simulated as mixture of beta densities, also known as Bernstein polynomials, as follows:

$$\begin{gathered} p_i(x) = \sum_{j=1}^{K} w_{ij} \beta(x; j, K-j), \\ \boldsymbol{w}_i \sim \text{Dirichlet}_K(0.01). \end{gathered} \tag{11.26}$$

Where $\beta(x; a, b)$ denotes the density of a beta distributed random variable with parameters (a, b) evaluated in x. By definition, the p_is generated from (11.26) have a fixed support $I = [0, 1]$. See Figure 11.C.1.

In this setting instead, we let μ_i in (11.20) be the probability measure associated to p_i and not its smoothed version. Hence, in addition to the amount of information lost during the PCA another factor comes into play: the amount of information that is lost due to the B-spline representation.

Figure 11.C.2 shows the results. We can see that the reconstruction errors decrease when the dimension of the principal component increases both for the simplicial and

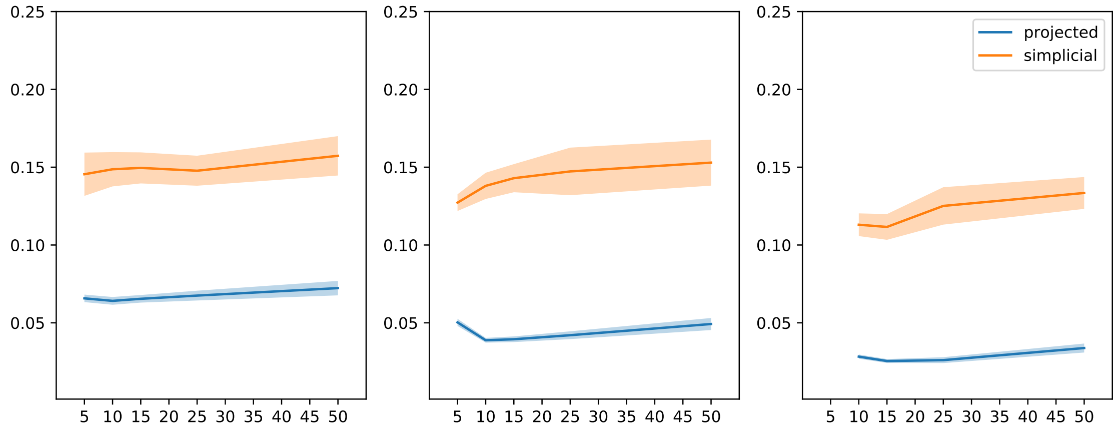


Figure 11.C.2: Results for the third scenario. All the panels show the reconstruction error as a function of the number of the spline basis functions. From left to right the results are obtained using the 2, 5 and 10 dimensional PCA. The solid lines represent the mean of 10 independent runs on independent data sets from (11.26) and the shaded area represent ± one standard deviation.

projected PCA. Moreover, as the number of B-spline basis increase, the performance tends to get a little bit worse for both the approaches. We believe that this is due to an increased variance in the B-spline estimation of the quantile functions and (clr of) pdfs. In fact, computing the spline approximation for a single function amounts to solving a linear regression problem and increasing the dimension of the B-spline basis corresponds to increasing the number of regressors. Hence, letting B the matrix with columns $\psi_1, \ldots, \psi_J$ (evaluated on a grid), the variance of the OLS estimate of the coefficients $\boldsymbol{a}$ is proportional to $(B^T B)^{-1}$. When increasing the number of B-splines, the entries in $B^T B$ become closer to zero, since the support of each of the spline basis becomes smaller. This leads to smaller precision (and higher variance) in the estimator for $\boldsymbol{a}$.

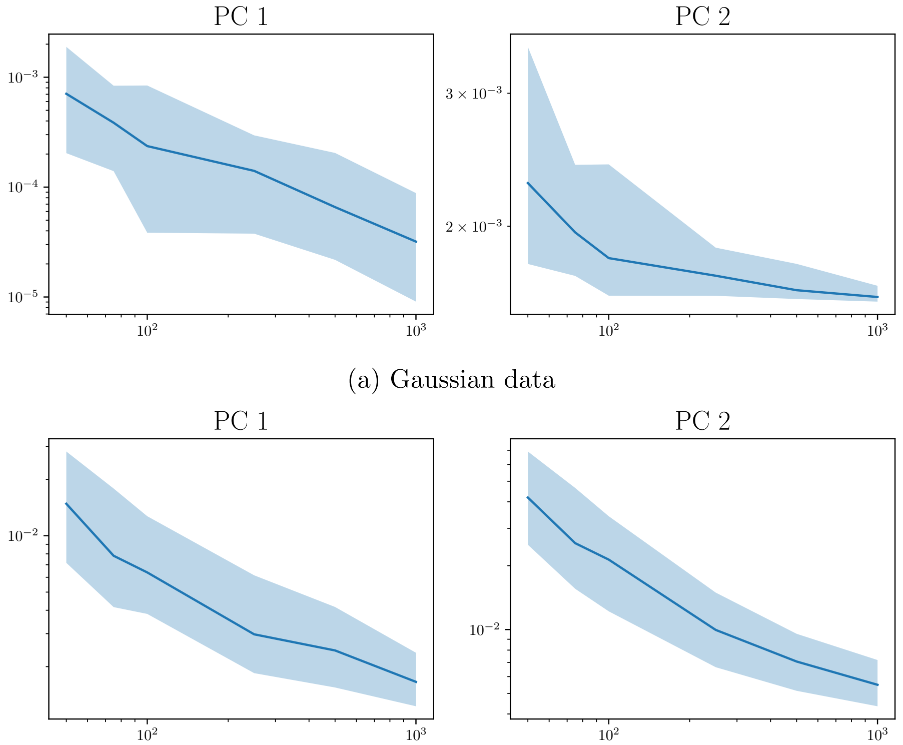
Another interesting thing to notice is that the simplicial PCA exhibits a much larger variance in the reconstruction error. This is possibly due to the different degree of smoothness of the quantile functions and of the pdfs. As the quantile functions are smoother than the pdfs, their B-spline basis expansion should have lower variance and be more similar to the true quantiles.

11.C.2 Empirical Verification of Consistency Results and Choosing J

In this section, we provide additional simulations to verify the consistency results established in Section 11.6.

For the PCA, we consider the two data generating processes in equations (11.19) (Gaussian) and (11.21) (DPM). First, first we fix $J = 20$ spline basis (as we do throughout Section 11.7) and let n increase. Then, we also let J increase linearly with n. We estimate the 'true' principal directions by simulating 10^5 observations and using 2500 elements in the B-spline basis. Then, for any choice of n and J we generate another data set and compute the corresponding first two principal directions via the projected PCA and compute the L_2 norm between the 'true' directions and the estimated ones.

Figure 11.C.3 shows the case of fixed J for both data generation strategies. It is clear that in both cases the error quickly decreases to zero (observe that both the x and y axes are in log scale), but the convergence speed is surely sub-exponential when looking, for instance, at the second principal direction.



(a) Gaussian data

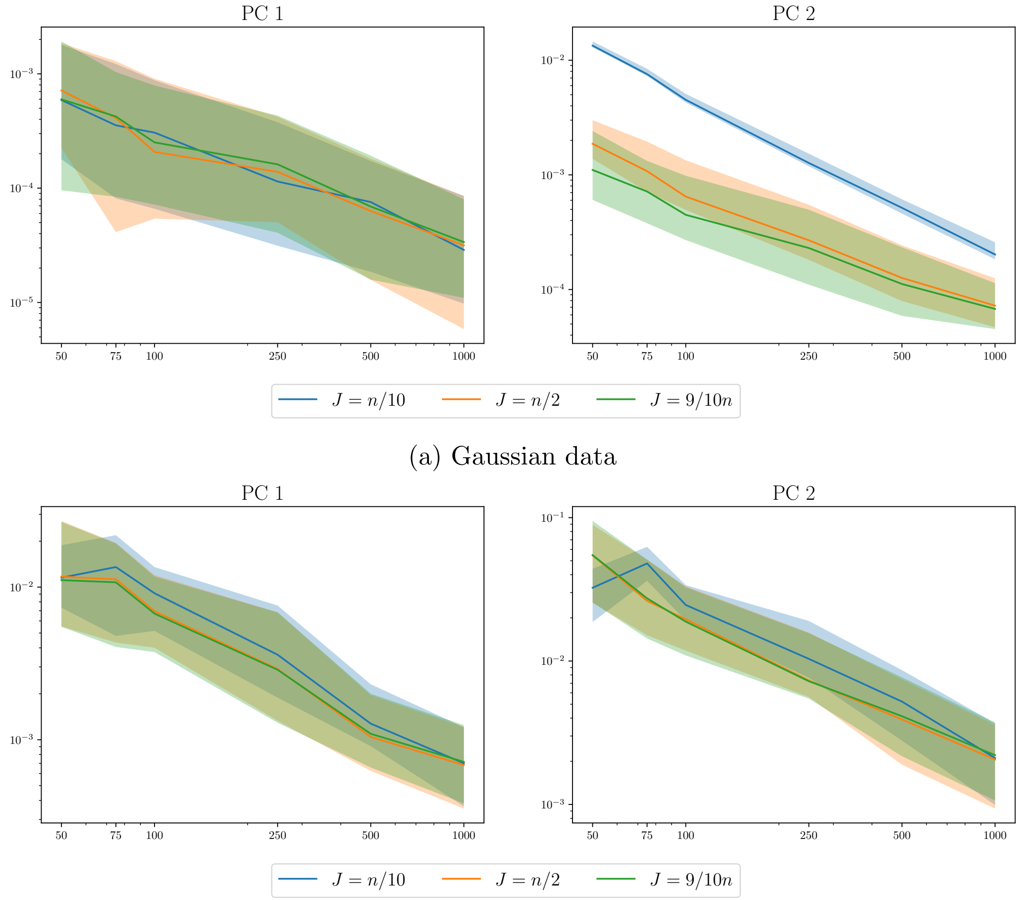
(b) DPM data

Figure 11.C.3: L_2 distance between estimated and true principal directions when $J = 20$ as a function of n. Solid line represents the median and the shaded area to a 90% confidence interval estimated from 100 independent repetition.

When increasing the number of basis elements with n, we consider three strategies letting $J = n/10$, $n/2$ and $9/10n$ respectively (rounded to the closest integer). Figure 11.C.4 shows the errors between the true and estimated principal directions in this case. Note that the convergence rate looks exponential for both data generating processes for every choice of $J = J(n)$ (increasing with n). In the case of Gaussian data, we observe smaller errors (as low as 10^{-5} for the first direction and 10^{-4} for the second direction) than in the case of the more challenging DPM data set, see Figure 11.C.4. For the former data set, using a large number of basis functions such as $9/10n$ or $n/2$ provides a much better fit than using $n/10$ basis functions on the second principal direction. For DPM data, the errors are in general two orders of magnitude higher than with Gaussian data. This is likely due to the different data generating process, which results in a more challenging problem. Interestingly, the errors are almost equal for all values of J (when fixing n).

Let us now analyze the projected regression. The independent variable are generated similarly to Section 11.8, by discretizing the interval [0, 1] in 1,000 equispaced intervals, the value of the quantile function $F^{-}_{z_i}$ in the j-th interval equals $\sum_{k=1}^{j} \delta_{ik}$ and $(\delta_{i1}, \ldots, \delta_{i1000}) \sim \text{Dirichlet}(0.01, \ldots, 0.01) + \mathcal{U}([0, 5])$. We fix the kernel $\beta^{\star}(t, s)$ (details are given below) and let quantile functions $F^{yi} = \Pi_{L_2([0,1]^{\uparrow})} \circ \Gamma_{\beta^{\star}}(F^{-}_{zi}) + \mathcal{N}(0, (0.1)^2)$.

We consider two different choices of $\beta^{\star}$: a smooth function $\beta^{\star}_1(t, s) = (t - 1/2)^3 + (s - 1/2)^3$, for which we expect that a small number of spline basis will give a low error, and



(a) Gaussian data

(b) DPM data

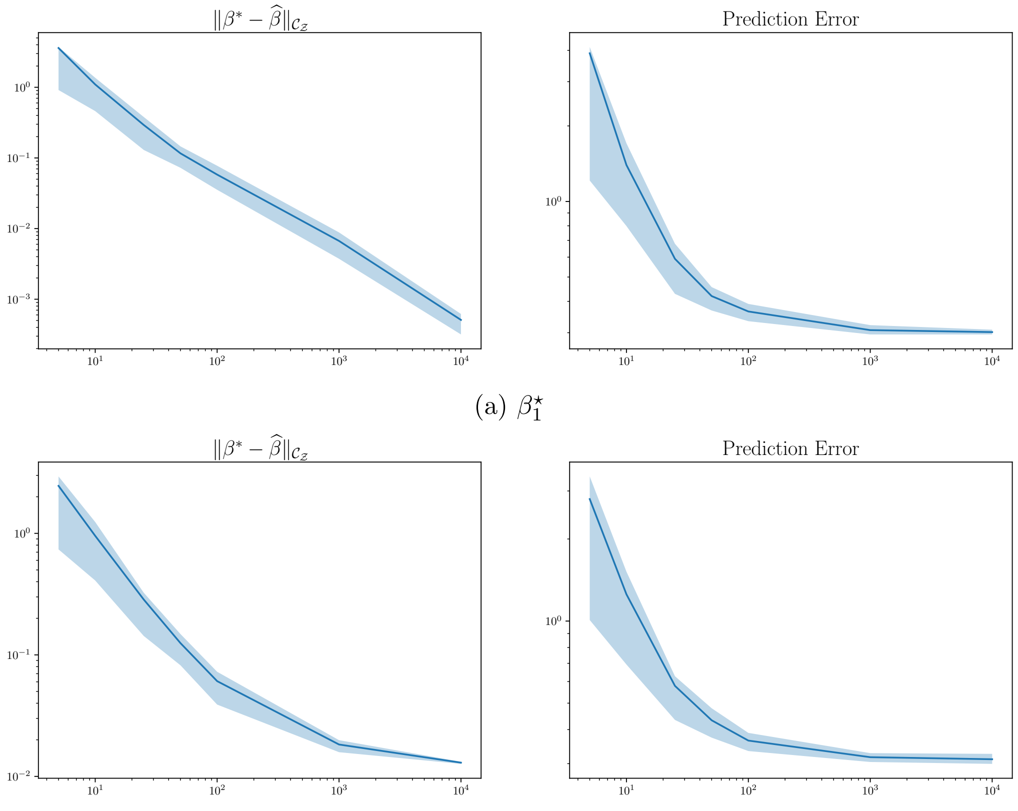
Figure 11.C.4: L_2 distance between estimated and true principal directions as a function of n for different choices of J. Solid line represents the median and the shaded area to a 90% confidence interval estimated from 100 independent repetition.

a rougher function $\beta_2^\star(t,s)$ defined as

$$\beta_2^\star(t,s) = \sum_{k,h=1}^{10} \beta_1^\star(0.1k, 0.1h)\mathbb{I}[(t,s) \in [0.1(k-1), 0.1k) \times [0.1(h-1), 0.1h)]$$

that is, $\beta_2^\star$ corresponds to an approximation of $\beta_1^\star$ on a 10×10 grid. As in the case of PCA, we present two simulations for each choice of $\beta_i^\star$, i=1,2, where we first fix the number of spline basis $J = 20$ while increasing the sample size n and second compare the performance for various values of J. We do not adopt the same strategy of setting J as a fraction of the number of n since the number of parameters to estimates grows quadratically with J which makes the computational cost substantial when $J \geq 100$. We measure both the seminorm error $\|\widehat{\beta} - \beta^\star\|_{\mathcal{C}_\mathcal{Z}}$ and the mean square prediction error on an unseen 'test' set of $1,000$ samples.

Figure 11.C.5 shows the seminorm error and the prediction error when $J = 20$ as n increases, while in Figure 11.C.6 various values of J are also considered. When data are generated from $\beta_1^\star$, $J = 20$ spline basis is more than enough (and actually $J = 10$ would suffice) and the seminorm error in Figure 11.C.5(a) and Figure 11.C.6(a) decays exponentially while the prediction error reaches the irreducible error with $n = 10^3$ samples. When data are generated from $\beta_2^\star$ the seminorm error does not show the same exponential



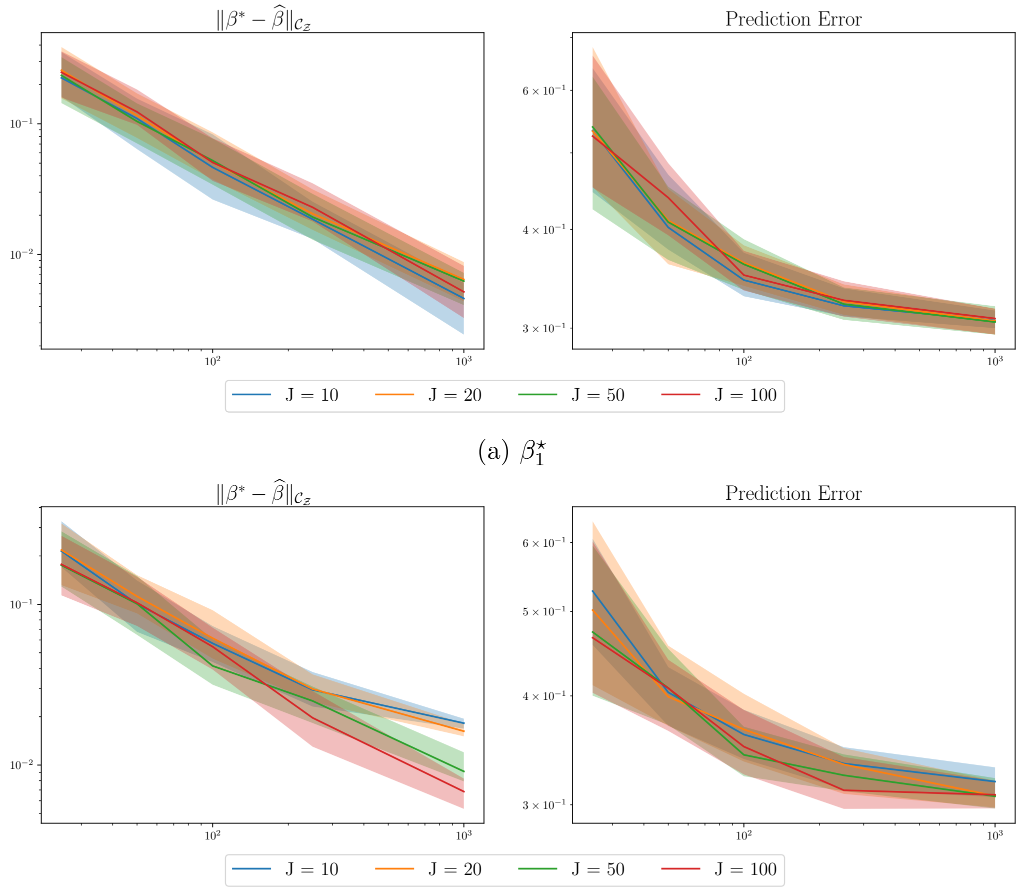
(a) $\beta_1^\star$

(b) $\beta_2^\star$

Figure 11.C.5: Seminorm error (left) and mean square prediction error (right) for different choices of the kernel used to generate data, when $J = 20$ as a function of n. Solid line represents the median and the shaded area to a 90% confidence interval estimated from 100 independent repetition.

decay when $J = 20$ (see Figure 11.C.5(b)), but it does for larger values of J, in particular it seems that the error obtained with $J = 50$ is the same obtained when $J = 100$, see Figure 11.C.6(b). Hence, it is clear that the choice of J is crucial to obtain a fast decay of the error: when the kernel to be approximated is not very smooth, a larger values of spline basis elements are needed, as one would expect.

We conclude this discussion by giving a practical advice on how to select J for a given data set. Our suggestion is to let J to be the smallest value that allows for a reconstruction error smaller than a given threshold, which may depend on the specific inferential task. For instance, if the problem is PCA and the goal is to provide a descriptive analysis of the variability, a (relative) approximation error below 0.05 will typically give satisfactory results. If instead the goal is only to perform dimensionality reduction and working on the scores of a PCA as Euclidean data, one should aim for a lower approximation error, possibly of the order of 10^{-4}. A similar reasoning can be applied to the regression: if the goal is mainly to interpret the estimate $\widehat{\beta}$ a larger reconstruction error can be allowed. If instead one is interested in obtaining very accurate predictions, a lower error is preferred. For instance, when $\beta_1^\star$ is used to generate the data, the reconstruction error for both dependent and independent variables is below 10^{-4} for $J \geq 20$, while to get to the same error when $\beta_2^\star$ is used one must use $J = 100$ basis.



(a) $\beta_1^\star$

(b) $\beta_2^\star$

Figure 11.C.6: Seminorm error (left) and mean square prediction error (right) for different choices of the kernel used to generate data, as a function of n for different values of J. Solid line represents the median and the shaded area to a 90% confidence interval estimated from 100 independent repetition.

11.D Limitations of the projected framework

11.D.1 When the projected PCA performs poorly

Here, we show an example to highlight the limitations of the proposed framework, specifically of the projected PCA. The main idea behind this example is that the projected principal directions will be different from the nested geodesic ones when data are concentrated around the 'borders' of X, as in the trivial example shown in Figure 11.3.1. In the Wasserstein case, X is the space of quantile functions so that the border composed of functions that are constant on a subset of $[0, 1]$.

Hence, we consider the following data generating process, modeling directly the quantile functions

$$F_i^-(t) = \begin{cases} v_{i1}, & \text{if } t < 0.5 \\ v_{i1} + v_{i2}, & \text{if } t > 0.5 \end{cases}$$

where $v_{ij} \sim \max\{0, \mathcal{N}(0, 1)\}$ independently. See Figure 11.D.1 for a random sample from this data generating process.

In this case, computing the projected PCA results in an interpretability score IS_k equal to one for $k = 1, 2$ and equal to zero for $k = 3, 4, \ldots$. Hence, from the third

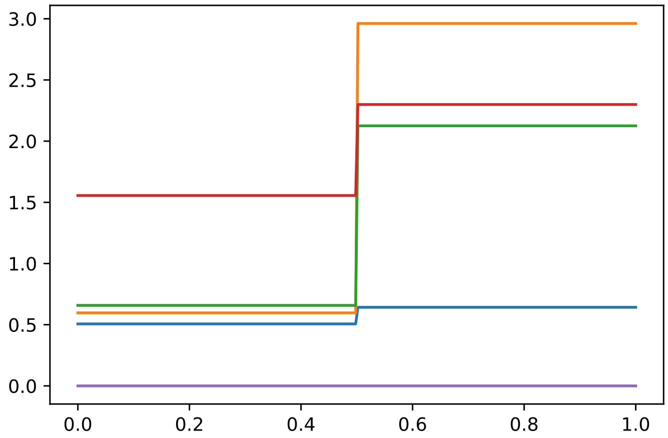


Figure 11.D.1: Five quantile functions from the data generating process considered in Appendix 11.D.1

principal direction onward, the projected PCA does not give any reliable information and, if those directions are needed, in this case a nested PCA could be preferred. Despite the poor interpretability scores from the third direction onward, the reconstruction errors are always good as $NRE_1 = 0.26$ and $NRE_k \approx 10^{-6}$ for $k \geq 2$. Moreover, the ghost variances GV_k are smaller than 10^{-10} for all values of k, so that this particular data set would be a good candidate for the hybrid methods mentioned in Section 11.7.2.

In summary, in our experience, the performance of the projected PCA can suffer when considering the interpretability of the directions associated to lower variability, but usually (at least always in our examples) gives a reasonable reconstruction error and ghost variance.

11.D.2 Inconsistent scores when increasing dimensions

Here, we highlight a feature which is shared by both projected and nested PCA, that is, the scores of the projection onto a projected principal component are dependent on the dimension of the principal component, as already noted in Section 11.3.1.

This can be considered a limitation to those frameworks because it contributes to the complexity of the analysis: one has always to fix the dimension of the chosen principal component and use the scores accordingly obtained. For instance, the scores, both for nested and projected PCAs, coincide with the L_2 scores when the dimension of the principal components is equal to the cardinality of the spline basis J. This happens because the principal components are not linear subspaces. As a consequence also the interpretability score of a direction is dimension-dependent.

Hence, the choice of the dimension k must be carried out balancing (i) a parsimonious representation, (ii) a low reconstruction error, so that the projections on the principal components yield good approximations of the data, and (iii) the intepretability score of the directions.

Thus, opposed to standard Euclidean PCA, where the $k+1$-th direction does not change the behavior of the data along the previous k directions (i.e., the scores), when doing (any) PCA in Wasserstein space the whole picture must always be taken into account, both for nested and projected PCA to assess the interpretability of the results.

Finally, note that such interpretability might be low for both intrinsic and extrinsic methods, but this means that the Wasserstein metric may not be the most adequate to capture and explain the variability of the data set.

12. Circular Wasserstein PCA

In this chapter, we extend the Wasserstein PCA in Chapter 11 to measures supported on $\mathbb{S}_1 := \{(x, y) \in \mathbb{R}^2 : x^2+y^2 = 1\}$. To this end, we provide a detailed characterization of the Wasserstein space for measures on $\mathbb{S}_1$, giving explicit formulas for the optimal transport maps as well as several characterizations related to the weak Riemannian structure of the Wasserstein space. We propose a convex-log PCA where we first map all the data to the tangent space at the barycenter and then solve a PCA problem in a convex cone. We discuss a numerical algorithm to approximate the Wasserstein barycenter and validate it empirically. A theoretical proof of its convergence remains an open and interesting problem, which motivates the study of differential calculus in the Wasserstein space.

12.1 Introduction

The development and progression of optic neuropathies, such as glaucoma, are often associated with a neuroretinal rim (NRR) thinning of the optic nerve head. In Ali et al. (2021) a data set of high resolution circular measurements based on optical coherence tomography (OCT) on NRR phenotypes is presented, arguing that baseline structural heterogeneity in the eyes can play a key role in the progression of optic neuropathies. The OCT produces a circular scan of the eye measuring NRR thickness. Therefore, each OCT can be considered as a function $f : \mathbb{S}_1 \to \mathbb{R}_+$, where $\mathbb{S}_1$ denotes the unit circle in $\mathbb{R}^2$, $\mathbb{S}_1 := \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\}$. Since the clinical interest is in the shape of the OCT rather than in the magnitude, it is standard practice to normalize the functions so that they can be seen as probability density functions. Therefore, comparing different OCTs is a problem of distributional data analysis.

The Wasserstein distance offers a natural framework for comparing probability measures, as witnessed by its popularity in very different fields. See, for instance, Bassetti et al. (2006), Bernton et al. (2019), Catalano et al. (2021) for statistical properties of the Wasserstein distance, Cao et al. (2019), Cuturi et al. (2019) and Cuturi and Doucet (2014) for applications in the field of machine and deep learning, Bernton et al. (2019) and Srivastava et al. (2015a) for applications in Bayesian computation. Different definitions of PCA (and related algorithms) for distributions under the Wasserstein metric have been proposed in Bigot et al. (2017), Cazelles et al. (2018) and Pegoraro and Beraha (2022). In these works, the space of square-integrable probability measures, endowed with the 2-Wasserstein metric (also called the Wasserstein space), is considered as a "Riemannian" manifold, and the characterization of the tangent space at an absolutely continuous probability measure (Ambrosio et al., 2008) is exploited to perform statistical analysis in a subset of a suitably defined subset of an L_2 space.

In particular, the *geodesic*-PCA in Bigot et al. (2017) and *projected* one in Pegoraro and Beraha (2022) are based on the explicit knowledge of optimal transport maps from an absolutely continuous measure to any other measure on $\mathbb{R}$. These maps can be constructed by composing the cumulative distribution function of the starting measure with the quantile function of the target measure. The *log*-PCA in Cazelles et al. (2018) can be, in principle, applied to distributions over more complex domains. However, as discussed

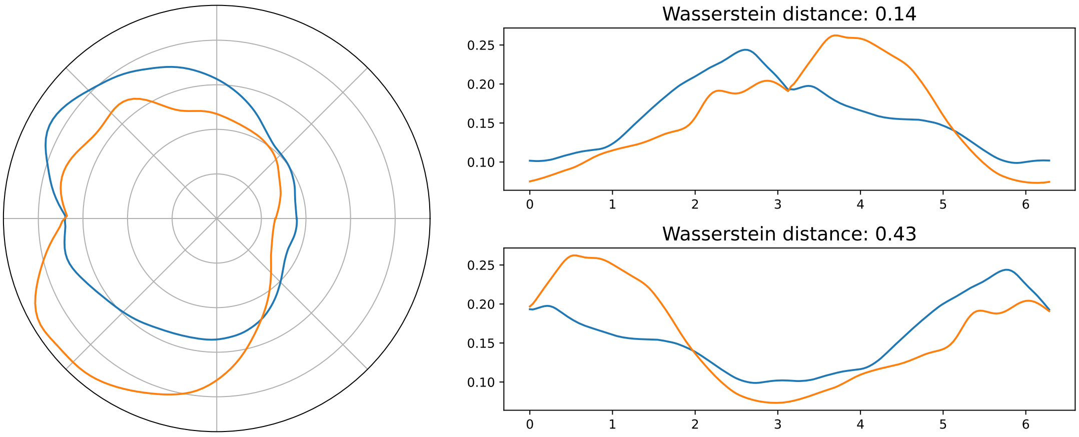


Figure 12.1.1: Two OCT samples on $\mathbb{S}_1$ (left) and when unrolled on $[0, 2\pi]$ (left) starting from 0 (top) or from $\pi/2$ (bottom) and the associated Wasserstein distances computed between the probability measures on $[0, 2\pi]$.

in Pegoraro and Beraha (2022), the *log*-PCA results in poor interpretability of the components and does not allow for a real dimensionality reduction, since it is not possible to work on the scores.

The main difficulty in extending the previously proposed approaches to measures on $\mathbb{S}_1$ is that the circle does not possess a natural ordering, unlike $\mathbb{R}$. Therefore, the concepts of the cumulative distribution function and the quantile function are not well defined. Starting from any point θ on the circle, we can "unroll" it and consider a bijection between $\mathbb{S}_1$ and $[0, 2\pi]$ (or $[0, 1]$) so that it might be tempting to treat the distributions on $\mathbb{S}_1$ as distributions on an interval of the real line. However, the Wasserstein metric is then dependent on the chosen θ, as shown, for example, in Figure 12.1.1. This is clearly understood since this approach does not consider the natural Riemannian metric on $\mathbb{S}_1$.

Optimal transport for measures on manifolds is an active area of research. The characterization of optimal transport maps between probability measures on manifolds has been established in McCann (2001) and exploited in Gigli (2011) to define the tangent space of the Wasserstein space at any measure. The definition of the tangent in Gigli (2011) is extremely general but abstract, as it involves the notion of *c*-concavity (see, e.g., Gigli, 2011) which does not translate in a handy representation of the functions in the tangent.

In this paper, we first build an alternative definition of the tangent space, specific to measures on $\mathbb{S}_1$. This might be considered to be of independent interest. Then we build on our definition of tangent space to define a suitable PCA for probability measures on $\mathbb{S}_1$ using the Wasserstein distance.

12.2 Background on Optimal Transport

In this section, we provide a brief account of optimal transport and the Wasserstein distance for measures on compact manifolds. See, e.g., Ambrosio et al. (2008) for a detailed treatment. The technical details are deferred to Appendix 12.A.

Riemannian Manifolds. Informally, one can think of an n-dimensional smooth manifold M as a set which locally behaves like a Euclidean space: it can be covered with a collection of open sets $(U_i)_{i\geq 1}$ for which there exist homeomorphisms $\varphi : U_i \to \varphi(U_i) \subset \mathbb{R}^n$, called coordinate chart, which satisfy some compatibility conditions. We may refer to $(U_i, \varphi(U_i))$ as a *local parametrization* of the manifold. A Riemannian manifold (M, g) of dimension n is a smooth manifold M endowed with an inner product $g = (g_x)_{x\in M}$ on the tangent space T_xM at each point $x \in M$. Its tangent bundle TM is defined as

$$TM := \coprod_{x\in M} T_xM = \bigcup_{x\in M} \{x\} \times T_xM. \tag{12.1}$$

Each T_xM is a vector space of dimension n. The tangent bundle is itself a smooth manifold of dimension $2n$ with a standard smooth structure. See Lee (2013b) for an introduction to Riemannian manifolds.

The *exponential* map at $z \in M$ denoted by $\exp_z : TM \to M$ allows us to map a tangent vector $v \in T_xM$ onto the manifold itself. Informally, $\exp_z(v)$ is the arrival point of the geodesic starting at z with the direction v traveling for a unit of time. The *logarithmic* map $\log_z : M \to TM$, where it is defined, satisfies $\exp_z \circ \log_z(x) = x$. The inner product g induces the volume measure ω, which is locally (i.e., on a chart (U, φ)) given by

$$\mathcal{L}_M(A) = \int_{\varphi(A)} |\det(g(\varphi^{-1}(x)))|^{1/2} d\mathcal{L}(x) \tag{12.2}$$

for any measurable $A \subset U$. See Appendix 12.A for measure-theoretical details.

Wasserstein space. To define the Wasserstein metric, denote by $\mathcal{P}(M)$ the space of probability measures on M and let $c : M \times M \to \mathbb{R}_+$ be a cost function. The p-Wasserstein distance between two probability measures on M, say μ and ν, is

$$W_p(\mu,\nu)^p = \min_{\gamma\in\Gamma(\mu,\nu)} \int_{M\times M} c(x,y)^p d\gamma(x,y), \qquad \mu,\nu \in \mathcal{P}(M) \tag{12.3}$$

where $\Gamma(\mu,\nu)$ is the set of all probability measures on $M \times M$ with marginals μ and ν. The existence of (at least one) optimal plan γ^o that attains the minimum in (12.3) is ensured if c is lower semicontinuous (Ambrosio et al., 2008). Definition (12.3) is due to Kantorovich and can be seen as the weak formulation of Monge's optimal transportation problem, i.e.

$$W_p(\mu,\nu)^p = \inf_{T:T\#\mu=\nu} \int_M c(x,T(x))^p d\mu(x)$$

where # denotes the pushforward operator: $T\#\mu(A) = \mu(T^{-1}(A))$ for all measurable A. It can be proven that when an optimal map exists, then this induces an optimal transport plan $\gamma^o = (\mathrm{Id}_M, T)\#\mu$ and the two formulations are equivalent. However, there are several situations in which Monge's problem has no solution.

In the following, we will always consider the Riemannian distance $\mathrm{d}_R(\cdot,\cdot)$ as cost function and set $p = 2$. We restrict our focus on measures in the 2-Wasserstein space, that is the subset of probability measures

$$\mathcal{W}_2(M) = \Big\{\mu \in \mathcal{P}(M) : \int_M \mathrm{d}_R(x,x_0)^2 d\mu(x) < \infty \text{ for every } x_0 \in M\Big\}.$$

This ensures that Wasserstein distance is always finite.

Geometry of the Wasserstein space. The Wasserstein space $(\mathcal{W}_2, W_2)$ can be endowed with a weak Riemannian structure induced by the tangent spaces of $\mathcal{W}_2$ at any absolutely continuous measure with respect to the volume measure (12.2). As in the case of measures supported in $\mathbb{R}^n$, the tangent spaces are subset of L^2 spaces of vector-valued functions defined on the ground space (in this case, M). Their definition needs some further background.

Consider a vector field $v : M \to TM$ such that for every $z \in M$, $v_z := v(z) \in T_zM$. To be more precise, denote by π the canonical projection map $\pi : TM \to M$, i.e. $\pi(z, v) = z \in M$, then v must be such that

$$\pi \circ v = \mathrm{Id}_M$$

where Id_M is the identity map on M. Let $S(M)$ be the collection of all such vector fields. Then, for a measure $\mu \in \mathcal{P}(M)$ we can define L^2_μ as

$$L^2_\mu(M) = \Big\{ v \in S(M) : \int g(v_z, v_z)^2 d\mu(z) < \infty \Big\}. \tag{12.4}$$

See Appendix 12.A for further details. For $v \in S(M)$ we can define the map $\exp(v) : M \to M$ such that $\exp(v)(z) := \exp_z(v_z)$ for $z \in M$. With this notation, we can state a fundamental theorem in optimal transportation due to McCann (2001).

Theorem 12.1 (Characterization of optimal transport plans)**.** *Let $\mu, \nu \in \mathcal{W}_2(M)$. If μ is absolutely continuous with respect to the volume measure* (12.2)*, there exists a unique optimal transport plan γ^o that has the form $\gamma^o = (Id_M, T)\#\mu$, where $T : M \to M$. Moreover, there exists a d^2_R-concave function ϕ such that $T = \exp(-\nabla\phi)$.*

The d^2_R-concavity condition is rather technical and not needed in the following, for this reason we report it only in Section 12.A, see Gigli (2011) for further details. To make explicit the dependence of the transport map on the source and target measures, we will use notation T^ν_μ to refer to the optimal transport map (OTM) from μ to ν.

The existence and uniqueness of optimal transport maps suggest the following definition of tangent spaces (Corollary 6.4 of Gigli, 2011)

$$\mathrm{Tan}_\mu(\mathcal{W}_2(M)) = \overline{\{v \in L^2_\mu(M) \mid \exists \varepsilon > 0 : (\mathrm{Id}_M, \exp(tv))\#\mu \text{ is optimal for } t \leq \varepsilon\}}^{L^2_\mu} \tag{12.5}$$

As in the case of Riemannian manifolds, we can define the exponential and logarithmic maps that allow to move from the tangent space $\mathrm{Tan}_\mu(\mathcal{W}_2(M))$ to the Wasserstein space and vice versa.

$$\begin{aligned} \exp_\mu &: L^2_\mu(M) \to \mathcal{W}_2(M), \qquad && \exp_\mu(v) = \exp(v)\#\mu \\ \log_\mu &: \mathcal{W}_2(M) \to L^2_\mu(M), \qquad && \log_\mu(\nu) = v \text{ s.t. } \exp(v) = T^\nu_\mu \end{aligned} \tag{12.6}$$

This structure is usually referred to as the *weak Riemannian structure* of $\mathcal{W}_2(M)$.

12.3 Optimal Transport on the Circumference

In this section, we specialize the general theory outlined in Section 12.2 to the case of measures supported on the unit-radius circumference.

12.3.1 Geometry of $\mathbb{S}_1$

For our purposes, it is convenient to define the unit-radius circumference as $\mathbb{S}_1 := \{z \in \mathbb{C} : |z| = 1\}$, where $|\cdot|$ denotes the module of a complex number. We first present the smooth (group) structure of $\mathbb{S}_1$ and then describe its Riemannian structure.

To endow $\mathbb{S}_1$ with a group structure, we start by considering the map $\exp_c : \mathbb{R} \to \mathbb{S}_1$ defined as $\exp_c(x) = e^{2\pi i x}$, and the map $\log_c : \mathbb{S}_1 \to \mathbb{R}$ defined as $\log_c(z) = x \in [0, 1)$ such that $z = e^{2\pi i x}$. Note that log_c is right inverse of $\exp_c$, i.e., $\exp_c \circ \log_c = \mathrm{Id}_{\mathbb{S}_1}$. The exponential map $\exp_c$ is usually referred to as *universal covering* of $\mathbb{S}_1$ (Munkres, 2000). Then, define the operation $\cdot : \mathbb{S}_1 \times \mathbb{S}_1 \to \mathbb{S}_1$ as $z \cdot w = \exp_c(\log_c(z) + \log_c(w))$. Informally speaking, $\log_c(z)$ is the "angle" associated with the polar representation of z and $\cdot$ is the sum of the angles. It can be trivially seen that $(\mathbb{S}_1, \cdot)$ is a group and $\exp_c : (\mathbb{R}, +) \to (\mathbb{S}_1, \cdot)$ is a group morphism.

Through $\exp_c$ and $\log_c$ we can define the smooth structure of $\mathbb{S}_1$ by considering at each $z \in \mathbb{S}_1$ the map $\exp_z(x) := \exp_c(x + \log_c(z))$, that is the shifted version of the exponential map, and $\log_z(w) = y$ such that $y \in [-1/2, 1/2)$ and $\exp_z(\log_z(w)) = w$. Letting $V_z := \mathbb{S}_1 \setminus \{-z\}$, we have that for each $z \in \mathbb{S}_1$ the couple $(V_z, \log_z)$ is a coordinate chart. With this differential structure $\mathbb{S}_1$ is a Lie group and its tangent bundle is $T\mathbb{S}_1 = \{(x, v) \,|\, x \in \mathbb{S}_1 \text{ and } v \in T_x\mathbb{S}_1\} \simeq \mathbb{S}_1 \times \mathbb{R}$. We call 1 the point $(1, 0)$ that gives the neutral element in $\mathbb{S}_1$.

We consider the Riemannian metric g is induced by the embedding $\mathbb{S}_1 \hookrightarrow \mathbb{C} \simeq \mathbb{R}^2$, that is $g_z(x, y) = xy$ for $x, y \in T_z\mathbb{S}_1 \simeq \mathbb{R}$. This induces the arc-length distance $\mathrm{d}_R(z, w) = |\log_c(z) - \log_c(w)|$. Note that $det(g) \equiv 1$, so that $\mathcal{L}_{\mathbb{S}_1} = \exp_c \#\mathcal{L}$ or, equivalenty, $\log_c \#\mathcal{L}_{\mathbb{S}_1} = \mathcal{L}$. Thus, for any $f : \mathbb{S}_1 \to \mathbb{R}$

$$\int_{\mathbb{S}_1} f(z) d\mathcal{L}_{\mathbb{S}_1}(z) = \int_{[-1/2,1/2)} f(\exp_c(x)) d\mathcal{L}(x) \tag{12.7}$$

For further details, see Appendix 12.A.

12.3.2 Optimal transport maps

With the notation introduced in the previous section, we now focus on the optimal transportation problem on $M = \mathbb{S}_1$ endowed with its Riemannian distance d_R.

The fundamental observation is that a measure μ on $\mathbb{S}_1$ can be equivalently represented by a *periodic* measure on $\mathbb{R}$ defined as $\widetilde{\mu}(A) := \mu(\exp_c(A))$ for measurable A, which entails $\widetilde{\mu}(A) = \widetilde{\mu}(A + p)$ for any $p \in \mathbb{Z}$, where $A + p$ amounts to shifting all the points in A by the amount p. Then we define the "periodic cumulative distribution function" associated with $\widetilde{\mu}$ as $F_{\widetilde{\mu}}(x) = \widetilde{\mu}([0, x))$ for $x \in [0, 1]$ and extend it over $\mathbb{R}$ via the rule $F_{\widetilde{\mu}}(x+1) = F_{\widetilde{\mu}}(x)+1$. For $\theta \in \mathbb{R}$, let $F^{\theta}_{\widetilde{\mu}}(x) = F_{\widetilde{\mu}}(x) + \theta$ denote a vertical shift of the cumulative distribution function. Note that the measure induced by $F^{\theta}_{\widetilde{\mu}}$ is independent from θ and is always $\widetilde{\mu}$. This easily follows from, for instance, $\widetilde{\mu}([a, b]) = F^{\theta}_{\widetilde{\mu}}(b) - F^{\theta}_{\widetilde{\mu}}(a) = F_{\widetilde{\mu}}(b) - F_{\widetilde{\mu}}(a)$.

Denote with $F^{-}_{\widetilde{\mu}}$ the associated quantile function, i.e., the (generalized) inverse of $F_{\widetilde{\mu}}$. We have that $(F^{\theta}_{\widetilde{\mu}})^{-}(x) = F^{-}_{\widetilde{\mu}}(x - \theta)$. If we restrict the quantiles on $[0, 1) = \log_c(\mathbb{S}_1)$, then θ acts as a rotation of the quantiles around the circle, by a factor of $z_{\theta}^{-1} = \exp_c(-\theta)$. Hence, the 0-th quantile $(F^{\theta}_{\widetilde{\mu}})^{-}(0)$ is not 0 but z_{θ}^{-1}. Equivalently, $F^{\theta}_{\widetilde{\mu}}(y) = \widetilde{\mu}([z_{\theta}^{-1}, y))$.

The following theorem provides an explicit characterization for the optimal transport maps between two measures on $\mathbb{S}_1$.

Theorem 12.2. *Define θ^* as the solution of the following minimization problem:*

$$\theta^* = \operatorname*{arg\,min}_{\theta \in \mathbb{R}} \int_0^1 \left(F^{-}_{\widetilde{\mu}}(u) - (F^{\theta}_{\widetilde{\nu}})^{-}(u) \right)^2 \mathrm{d}u \tag{12.8}$$

Then the optimal transport map between μ and ν is

$$T^{\nu}_{\mu} := \exp_c \circ \left((F^{\theta^*}_{\widetilde{\nu}})^{-} \circ F_{\widetilde{\mu}} \right) \circ \log_c . \tag{12.9}$$

Note that (12.9) is essentially identical to the expression of optimal transport maps for measures on $\mathbb{R}$. In that case, setting $\exp_c = \log_c = \mathrm{Id}$ and $\theta^* = 0$ we recover the classical formulation of OTMs for measures on the real line. In the following, we will write $\widetilde{T}^{\widetilde{\nu}}_{\widetilde{\mu}} := (F^{\theta^*}_{\widetilde{\nu}})^{-} \circ F_{\widetilde{\mu}}$ to denote the map between $\widetilde{\mu}$ and $\widetilde{\nu}$ associated with the optimal θ^* in (12.8). Although $\widetilde{T}^{\widetilde{\nu}}_{\widetilde{\mu}}$ is not "optimal" (since the cost associated to the transport of periodic measures is either zero or unbounded), we will refer to it as the optimal transport map between $\widetilde{\mu}$ and $\widetilde{\nu}$ in light with its connection with T_{μ}^{ν}.

Let us give some intuition behind the optimal transport map T_{μ}^{ν}. Observe that pre-composing $(F_{\widetilde{\nu}}^{\theta^*})^-$ with $\left(F_{\widetilde{\mu}}\right)_{|\,[0,1]}$, obtaining $\widetilde{T}_{\widetilde{\mu}}^{\widetilde{\nu}}$, means transporting quantiles identified by $F_{\widetilde{\mu}}^-$ onto the corresponding shifted quantiles of $(F_{\widetilde{\nu}}^{\theta^*})^-_{|\,[0,1]}$, in an anti-clockwise order (due to the definition of $\exp_c$). Note that $T_{\widetilde{\mu}}^{\widetilde{\nu}}((F_{\widetilde{\mu}})^-(0)) = T_{\widetilde{\mu}}^{\widetilde{\nu}}(0) = F_{\widetilde{\nu}}^-(-\theta^*) =: x_{-\theta^*}$ and

$$T_{\widetilde{\nu}}^{\widetilde{\mu}}((F_{\widetilde{\mu}})^-(1)) \leq T_{\widetilde{\nu}}^{\widetilde{\mu}}(1) = F_{\widetilde{\nu}}^-(1-\theta^*) = 1 + F_{\widetilde{\nu}}^-(-\theta^*) = 1 + x_{-\theta^*},$$

which means that the optimal transport maps sends $[0,1)$ into $[x_{-\theta^*}, 1 + x_{-\theta^*})$. As a consequence we can think at this situation as 'unrolling' the circle in two different points, namely $z_{\theta^*}^{-1} = \exp_c(-\theta^*)$ for ν and $1 = \exp_c(0)$ for μ, and then matching the measures induced on $\mathbb{R}$. For instance, suppose μ and ν have densities f_μ and f_ν with respect to the Lebesgue measure on $\mathbb{S}_1$, $\mathcal{L}_{\mathbb{S}_1}$, then $(F_{\widetilde{\nu}}^{\theta})^-_{|\,[0,1]}$ is the quantile function associated with the density $f_\nu(\exp_c(x))$ supported on $[x_{-\theta}, 1 + x_{-\theta}]$. Clearly no action is taken on μ and thus we transport $f_\mu(\exp_c(x))$ supported on $[0,1]$ onto $f_\nu(\exp_c(x))$ supported on $[x_{-\theta}, 1+x_{-\theta}]$. The parameter θ^* then selects the optimal point from which to start unrolling the circle for ν.

In later sections, we will develop statistical tools to analyze distributions on $\mathbb{S}_1$ based on the optimal transport maps T_i from a reference distribution to the i-th datapoint. Thus, it is essential to characterize the optimal transport maps on $\mathbb{S}_1$ in light of the associated maps $\widetilde{T}$ between periodic measures on $\mathbb{R}$.

Theorem 12.3. *Given μ a.c. measure and $\nu \in \mathcal{W}_2(\mathbb{S}_1)$, $\widetilde{T} := (F_{\widetilde{\nu}}^{\theta^*})^- \circ F_{\widetilde{\mu}}$ is an optimal transport map if and only if:*

$$\int_0^1 \widetilde{T}(u) - u \, \mathrm{d}u = 0. \tag{12.10}$$

12.3.3 Weak Riemannian structure

We now specialize the definition of $\mathrm{Tan}_\mu(\mathcal{W}_2(M))$ and the associated exponential and logarithmic maps when $M \equiv \mathbb{S}_1$. Furthermore, we establish properties of the logarithmic map that will be fundamental to develop a coherent statistical framework for analyzing probability measures in $\mathcal{W}_2(\mathbb{S}_1)$.

For our purposes, it is convenient to define $L^2_\mu(\mathbb{S}_1)$ as

$$\begin{aligned} L^2_\mu(\mathbb{S}_1) :&= \left\{ v : \mathbb{S}_1 \to \mathbb{R} \text{ such that } \int_{\mathbb{S}_1} v^2(x) d\mu(x) < +\infty \right\} \\ &= \left\{ v : [0,1) \to \mathbb{R} \text{ such that } \int_0^1 v^2(x) d\widetilde{\mu}(x) < +\infty \right\} \end{aligned}$$

where the second equality follows, with a slight abuse of notation, by considering $v \mapsto v \circ \log_c$. Observe that we recover the space in (12.4) by identifying $v(x)$ as an element of $T_x\mathbb{S}_1$. Then, if μ is an absolutely continuous measure, we have

$$\mathrm{Tan}_\mu(\mathcal{W}_2(\mathbb{S}_1)) = \overline{\{v : L^2_\mu(\mathbb{S}_1) \,|\, \exists \varepsilon > 0 : (\mathrm{Id}_{\mathbb{S}_1}, \exp(tv)) \# \mu \text{ is optimal for } t \leq \varepsilon\}}^{L^2_\mu} \tag{12.11}$$

where we can interpret v as a function defined on $\mathbb{S}_1$ or $[0,1)$ according to our needs.

Note that the optimality condition in (12.11) is equivalent to saying that there exist ν such that $\exp(tv)$ is an optimal transport map between μ and ν. Then, by Theorem 12.2 and the fact that $\exp_z(v_z) = \exp_c(\log_c(z) + v_z)$, the vector field v in (12.11) can be written as $tv(\log_c(x)) = \widetilde{T}(x) - x$, where $\widetilde{T}$ is as in Theorem 12.2, so that the OTM is

$\exp_c(x + (\widetilde{T}(x) - x)) \equiv \exp_c(\widetilde{T}(x))$. Hence, we can restate the definition of tangent space in terms of the maps $\widetilde{T}$ as:

$$\mathrm{Tan}_\mu(\mathcal{W}_2(\mathbb{S}_1)) = \overline{\{\widetilde{T} : L^2_{\widetilde{\mu}}([0,1]) \,|\, \exists \varepsilon > 0 : \exp_c(\mathrm{Id} + t(\widetilde{T} - \mathrm{Id})) \text{ is OTM for } t \leq \varepsilon\}}^{L^2_\mu} \tag{12.12}$$

The definition of exponential and logarithmic map comes quite naturally:

$$\begin{aligned} \exp_\mu : L^2_\mu(\mathbb{S}_1) \to \mathcal{W}_2(\mathbb{S}_1), \qquad & \exp_\mu(\widetilde{T}) = \exp_c \circ \widetilde{T} \circ \log_c \#\mu \\ \log_\mu : \mathcal{W}_2(\mathbb{S}_1) \to L^2_\mu(\mathbb{S}_1), \qquad & \log_\mu(\nu) = \widetilde{T} \text{ s.t. } \widetilde{T}(x) = F^-_{\widetilde{\nu}}(F_{\widetilde{\mu}}(x) - \theta^*) \end{aligned} \tag{12.13}$$

where θ^* in the definition of the $\log_\mu$ map is as in Theorem 12.2. Observe that then $\exp_c \circ \widetilde{T} \circ \log_c$ is an OTM between μ and ν. Furthermore, from Theorem 12.3 we note that the vector field $v : [0,1) \to \mathbb{R}$ induced by an optimal transport map $\widetilde{T}$ ($v(u) = \widetilde{T}(u) - u$) satisfying (12.10) has zero mean when integrated along $\mathbb{S}_1$ with respect to $\mathcal{L}_{\mathbb{S}_1}$. In particular, note that this condition does not depend on μ and gives a purely geometric characterization of optimal transport maps. This is in accordance to other typically used optimality conditions such as cyclical monotonicity of the support of the transport plan and Brenier's characterization of OTMs for measures on $\mathbb{R}^n$ (Ambrosio et al., 2008).

We now provide some further characterizations of the optimal transport maps. These will be useful to investigate the map $\log_\mu$ and implementation of numerical algorithms.

Theorem 12.4. *Given μ a.c. measure, $\widetilde{T} : \mathbb{R} \to \mathbb{R}$ induces an optimal transport map between μ and $\nu := \exp_c \circ \widetilde{T} \circ \log_c \#\mu$ if and only if*

- *$\widetilde{T}$ is monotonically nondecreasing with $\widetilde{T}(x + p) = \widetilde{T}(x) + p$ for all $p \in \mathbb{Z}$*
- *$\widetilde{T}$ satisfies* (12.10)
- *$|\widetilde{T}(x) - x| < 1/2$ μ-a.e.*

From the previous result, it is immediate to prove

Corollary 12.1. *Let μ be an a.c. measure on $\mathbb{S}_1$. Then the image of* $\log_\mu$ *defined in* (12.13) *is a convex set.*

Moreover, the following proposition establishes the continuity of both $\exp_\mu$ and $\log_\mu$

Theorem 12.5. *Let μ be an a.c. measure on $\mathbb{S}_1$. Then*

1. *for any $\nu_1, \nu_2 \in \mathcal{W}(\mathbb{S}_1)$*

$$W_2^2(\nu_1, \nu_2) \leq \int_{\mathbb{S}_1} d_R^2(T_\mu^{\nu_1}, T_\mu^{\nu_2}) d\mu \leq \| \log_\mu(\nu_1) - log_\mu(\nu_2) \|^2_{L^2_\mu} .$$

In particular, the $\exp_\mu$ *map is continuous.*

2. *If $W_2(\nu, \nu_n) \to 0$ in $\mathcal{W}_2(\mathbb{S}_1)$ then*

$$\| \log_\mu(\nu_n) - \log_\mu(\nu) \|_{L^2_\mu} \to 0$$

3. *Let σ be an a.c. measure and $\{\mu_t\}_t$ be a sequence of a.c. measures such that $\mu_t \to \mu_0$ (in the Wasserstein metric) as $t \to 0$. Further assume that the support of σ and μ_t is convex and their density is bounded from above and strictly greater than zero. Then there exists $K > 0$*

$$\|\widetilde{T}_\sigma^{\mu_t} - \widetilde{T}_\sigma^{\mu_0}\| \leq K W_2(\mu_0, \mu_t)$$

12.4 PCA for Measures on $\mathbb{S}_1$

In this section, we demonstrate how the results obtained in Section 12.3 can be leveraged to develop a principal component analysis framework for measures on $\mathbb{S}_1$, by considering $\mu_1, \ldots, \mu_n \in \mathcal{W}_2(\mathbb{S}_1)$ as points of a "Riemannian manifold", cf. Section 12.3.3. This parallelism was first exploited to perform inference on the Wasserstein space in Bigot et al. (2017); Cazelles et al. (2018) to develop a PCA for probability measures on the real line, and later in Chen et al. (2021) and Zhang et al. (2020) who propose linear regression and autoregressive models for measures on $\mathbb{R}$ respectively.

The statistical techniques developed for manifold-valued data are typically in divided in *extrisinc* and *intrinsic* ones. The extrinsic approach consists of finding a linear space that approximates the manifold (or the region of the manifold where data are located), and perform inference on the projection of data onto the linear space, applying standard techniques developed for multivariate data in Euclidean spaces. Usually, such a linear space is the tangent space at the barycentric (mean) point. In the intrinsic case instead, the geodesic structure of the manifold is exploited to define a PCA based on the distance between datapoints and convex subsets of the manifold, whereby one considers convex subsets as the natural generalization of linear subspaces. Extrinsic techniques introduce an approximation that might significantly impact the results if the manifold is not well approximated. On the other hand, intrinsic techniques are usually computationally intensive and not suitable to analyze large datasets.

The weak Riemannian structure of the Wasserstein space allows us to define both intrinsic and extrinsic techniques as done in Bigot et al. (2017); Cazelles et al. (2018); Chen et al. (2021); Zhang et al. (2020); Pegoraro and Beraha (2022) for measures on the real line. In the previous papers, the intrinsic methods exploited the well-know isometry between $\mathcal{W}_2(\mathbb{R})$ and the "space of quantiles", that is the subset of L_2 made of monotonically non decreasing functions, so that $\mathcal{W}_2(\mathbb{R})$ can be seen as a convex cone inside a Hilbert space. Thus, intrinsic methods simply need to take into account the "cone constraints" (Pegoraro and Beraha, 2022). In the case of $\mathcal{W}(\mathbb{S}_1)$, there is no such isometry. Therefore, developing intrinsic methods would require to work with curves of probability measures. While we believe that the results established in Section 12.3 could be a first building block of such intrinsic methods. However, the continuity result in item (3.) of Theorem 12.5 suggests that the approximation we make when mapping data to the tangent space is not too coarse. The numerical illustrations presented in Section 12.5 seem to validate this claim. Thus, it might be the case that intrinsic methods, at least for PCA, would be essentially identical to extrinsic ones.

12.4.1 Log Convex PCA on $\mathcal{W}_2(\mathbb{S}_1)$

In the following, we will describe an *extrinsic* PCA for probability measures on $\mathbb{S}_1$. As shown in Corollary 6.6 of Gigli (2011), the tangent space at absolutely continuous measures is Hilbert so that we could apply standard PCA techniques to $\log_{\bar{\mu}}(\mu_1), \ldots, \log_{\bar{\mu}}(\mu_n)$, for some fixed measure $\bar{\mu}$. We call this approach "naive" log-PCA. However, as argued in Pegoraro and Beraha (2022), disregarding the fact that the image of the $\log_{\bar{\mu}}$ map is not the whole $\mathrm{Tan}_{\bar{\mu}}(\mathcal{W}_2(\mathbb{S}_1))$ tangent space, but only a convex subset, might produce misleading results. In particular, when two elements of the tangent space lie outside the image of $\log_{\bar{\mu}}$, returning to the Wasserstein space and then back to the tangent via $\log_{\bar{\mu}} \circ \exp_{\bar{\mu}}$ in general does not preserve distances or angles. This fact undermines, for instance, the interpretability of scores and principal directions when they lie outside $\log_{\bar{\mu}}(\mathcal{W}_2(\mathbb{S}_1))$: directions may not the orthogonal and variance inside $\mathcal{W}_2(\mathbb{S}_1)$ may not be decomposed appropriately.

To avoid the problems with the "naive" log-PCA, we propose the following definition

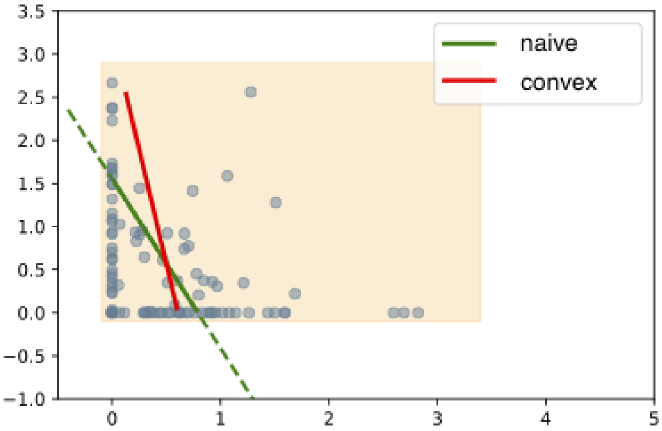


Figure 12.4.1: First principal direction found by the naive L_2 and the convex PCA when the space $H = \mathbb{R}^2$ and X is the yellow rectangle. The blue dots denote observations.

of log convex PCA, which amounts to performing a convex PCA (Bigot et al., 2017), thus taking into account the image of the log map, in the tangent space. Let us introduce some notation first. Let $X := \log_{\bar{\mu}}(\mathcal{W}_2(\mathbb{S}_1))$, $H := \mathrm{Tan}_{\bar{\mu}}(\mathcal{W}_2(\mathbb{S}_1))$. For a closed convex set $C \subset X$ and a point $x \in X$ let $d(x, C) = \arg\min_{y \in C} \|x - y\|_{L^2_{\bar{\mu}}}$ Let Sp denote the span of a set of vectors and $\mathcal{C}_{x_0}(U) := (x_0 + Sp(U)) \cap X$ for $x_0 \in X$ and $U \subset H$.

Definition 1. *Consider a collection of probability measures $\bar{\mu}, \mu_0, \mu_1, \ldots, \mu_n \in \mathcal{W}(\mathbb{S}_1)$. Let $\widetilde{T}_i = \log_{\bar{\mu}}(\mu_i) = \widetilde{T}_{\bar{\mu}}^{\mu_i}$, $i = 0, \ldots, n$. A $(k, \bar{\mu}, \mu_0)$ log convex principal component for $\mu_1, \ldots, \mu_n$ is the subset $C_k := \mathcal{C}_{\widetilde{T}_0}(\{w_1^*, \ldots, w_k^*\})$ such that*

1. *for $k = 1$,*

$$w_1^* = \operatorname*{arg\,min}_{w \in H, \|w\|=1} \sum_{i=1}^{n} d\left(\widetilde{T}_i, \mathcal{C}_{\widetilde{T}_0}(\{w\})\right)$$

2. *for $k > 1$,*

$$w_k^* = \operatorname*{arg\,min}_{w \in H, \|w\|=1, w \perp Sp(\{w_1^*, \ldots, w_{k-1}^*\}} \sum_{i=1}^{n} d\left(\widetilde{T}_i, \mathcal{C}_{\widetilde{T}_0}(\{w\})\right)$$

Figure 12.4.1 exemplifies the difference between the naive L_2 and the convex one in a simpler example when $H = \mathbb{R}^2$ and X is a convex subset. When data are close to the border of X, the L_2 metric between data and the principal components captures a variability that lies outside of the convex set. See also Pegoraro and Beraha (2022) for some indexes that quantify the loss of information of the L_2 PCA opposed to the convex one.

12.4.2 Computation of the Log Convex PCA via B-Spline approximation

The definition of convex PCA translates into a constrained optimization problem to find the directions $\{w_1^*, \ldots, w_k^*\}$. In Cazelles et al. (2018), the authors discretize the transport maps and solve the optimization problem via a forward-backward algorithm. As discussed in Pegoraro and Beraha (2022), a more efficient approach consists in approximating the transport maps via quadratic B-splines and solving a constrained optimization problem via an interior-point method. Here, we follow the second approach.

Let $\{\psi_1, \ldots, \psi_J\}$ a B-spline basis on equispaced knots in $[0, 1]$. We let $\widetilde{T}_i(x) \approx \sum_{j=1}^{J} a_{ij}\psi_j(x)$. Note that if the spline is quadratic then (i) the function $\sum_{j=1}^{J} a_j\psi_j(x)$ is monotonically nondecreasing if an only if the coefficients $a_1, \ldots, a_J$ are (see, e.g., Proposition 4 in Pegoraro and Beraha, 2022). Hence, from now on, we consider the ψ_j's to be quadratic spline basis functions on $[0, 1]$. The spline basis expansion also allows for faster computations of L_2 inner products: let E be a $J \times J$ matrix with entries $E_{i,j} = \int_0^1 \psi_i(x)\psi_j(x)\mathrm{d}x$ and $\boldsymbol{a}_i = (a_{i,1}, \ldots, a_{i,J})$, we have $\langle \widetilde{T}_i, \widetilde{T}_j \rangle = \langle \boldsymbol{a}_i, \boldsymbol{a}_j \rangle_E := \boldsymbol{a}_i^T E \boldsymbol{a}_j$. We denote by $\| \cdot \|_E$ the associated norm.

Similarly to Proposition 6 in Pegoraro and Beraha (2022), we obtain that the k-th direction $\boldsymbol{w}_k$ and the associated scores $\lambda^k_{1:n} = \lambda_1, \ldots, \lambda_n$ (of the observations the k-th direction) of the log-convex PCA can be computed by solving a constrained optimization problem. The objective function is clearly

$$\lambda^k_{1:n}, \boldsymbol{w}_k = \underset{\lambda_{1:n}, \boldsymbol{w}}{\arg\min} \sum_{i=1}^n \|\boldsymbol{a}_i - \boldsymbol{a}_0 - \sum_{j=1}^k \lambda_i^k \boldsymbol{w}_k\| \tag{12.14}$$

where $\lambda_i \in \mathbb{R}$ is the of score for the i-th datum along the k-th direction. Moreover, the usual orthogonality and unit-norm constraints must be satisfied:

$$\|\boldsymbol{w}\|_E = 1, \quad \langle \boldsymbol{w}_h, \boldsymbol{w} \rangle_E = 0, \quad h = 1, \ldots, k-1.$$

In addition to those, we must also require that $\sum w_j \psi_j$ belongs to $H := \mathrm{Tan}_{\bar{\mu}}(\mathcal{W}_2(\mathbb{S}_1))$. The monotonicity constraint is equivalent to

$$\lambda_i w_j + a_{0,j} - \lambda_i w_{j-1} - a_{0,j-1} \geq 0, \quad j = 2 \ldots J$$

that is the monotonicity of the spline coefficients since the splines are quadratics. Moreover, the "periodicity" constraint is satisfied by design. To impose (12.10), let $M_j = \int \psi_j(u)\mathrm{d}u$, then (12.10) is equivalent to

$$\sum w_j M_j = 1.$$

Finally, thanks to (12.10) it is sufficient to control the value of the function w at the initial point, i.e. $w_0 \in (-1/2, 1/2)$.

We implement the resulting constrained optimization problem using the `Python` package `pyomo` and approximate the solution using an interior point method using the Ipopt solver.

12.4.3 Wasserstein Barycenter

We are left to discuss the choice of the base point μ_0 of the PCA as well as the measure $\bar{\mu}$ at which the tangent space is considered. A natural candidate for both μ_0 and $\bar{\mu}$ is the (Wasserestein) barycenter, that is the Fréchet mean, which minimizes the Fréchet functional

$$F(\nu; \mu_1, \ldots, \mu_n) = \frac{1}{2n} \sum_{i=1}^n W_2^2(\nu, \mu_i). \tag{12.15}$$

Uniqueness of the Wasserstein barycenter has been studied in Agueh and Carlier (2011) in the case of measures supported on $\mathbb{R}^d$ and extended by Kim and Pass (2017) for measures on compact Riemannian manifolds. In particular, Theorem 3.1 in Kim and Pass (2017) establishes the uniqueness of the Wasserstein barycenter if at least one of the measures μ_j is absolutely continuous.

Numerical algorithms for computing the solution of (12.15) have been developed in Carlier et al. (2015); Srivastava et al. (2015b) for the case of atomic measures, whereby the optimization can be reduced to a linear program. Zemel and Panaretos (2019) instead propose an algorithm based on gradient descent which works for general measures on $\mathbb{R}^d$ (of which one must be absolutely continuous). In a nutshell, the gradient descent algorithm

Algorithm 3. Procrustes Barycenter

```
input Measures $\mu_1, \dots, \mu_n$, starting point $\nu$, threshold $\varepsilon$.
repeat
    Compute the optimal transport maps $\widetilde{T}_{\nu}^{\mu_i}$ as in Theorem 12.2.
    Set
        $\widetilde{\nu}' := \left(\frac{1}{n}\sum_{i=1}^{n} \widetilde{T}_{\bar{\mu}}^{\mu_i}\right) \# \widetilde{\nu}$
until $W_2(\nu, \nu') < \varepsilon$
Output $\bar{\mu} = \exp_c \circ (\widetilde{\nu}')$.
end
```

in Zemel and Panaretos (2019) starts from an initial guess of the barycenter and updates it by pushing forward the current guess ν_r of the barycenter the average of the transport maps between ν_r and all the measures. This procedure is guaranteed to converge to the barycenter under some technical conditions on the measures μ_i's. In particular, it converges in one iteration if the measures are *compatible* (see Section 2.3.2 in Panaretos and Zemel, 2020). As a drawback, this approach requires solving n optimal transportation problems at each iteration, which might be challenging outside the case of measures supported on $\mathbb{R}$ or location-scatter families, for which explicit solutions exist (Álvarez-Esteban et al., 2018). Taking a different approach, Cuturi and Doucet (2014) propose an approximate solution to the Fréchet mean by introducing in (12.15) an "entropic regularization" term, which makes optimization easier.

Here, we propose to use the gradient descent algorithm developed in Zemel and Panaretos (2019). Indeed, our Theorem 12.2 allows for explicit solutions to the optimal transportation problem. Moreover, as shown in Delon et al. (2010), the optimization problem in (12.8) is convex in θ so that finding θ^* is simple. We report the pseudocode for finding the barycenter in Algorithm 3.

We want to remark that we have not been able (yet) to prove the convergence of the algorithm to the barycenter. Note that this does not invalidate the results of the PCA. However, embedding the PCA in the tangent at the barycenter is to be preferred since, intuitively, this should result in the distance in the tangent space (at the barycenter) to be more similar to the distance in the Wasserstein space. In the following section we provide empirical evidence of its convergence, by comparing the output of Algorithm 3 to the one of the Sinkhorn algorithm proposed in Cuturi and Doucet (2014). From the technical point of view, the proofs in Zemel and Panaretos (2019) do not hold in our case, since they are based on sub-differentiability and super-differentiability results of the Wasserstein distance as provided in Theorems 10.2.2 and 10.2.6 in Ambrosio et al. (2008) which are stated for measures on separable Hilbert spaces.

12.5 Numerical Illustrations

12.5.1 Simulations for the Barycenter

Let us give an illustrative example of the peculiarities that may arise when considering distributions on $\mathbb{S}_1$. Consider the two measures on the leftmost panel in Figure 12.5.1. When the transport cost is the Euclidean one, the resulting barycenter is the one displayed in the rightmost panel: it has unimodal density with the same scale of the two measures and is centered exactly in the middle of them. When the cost instead is computed on $\mathbb{S}_1$, the barycenter becomes bimodal as shown in the middle panel of Figure 12.5.1. In

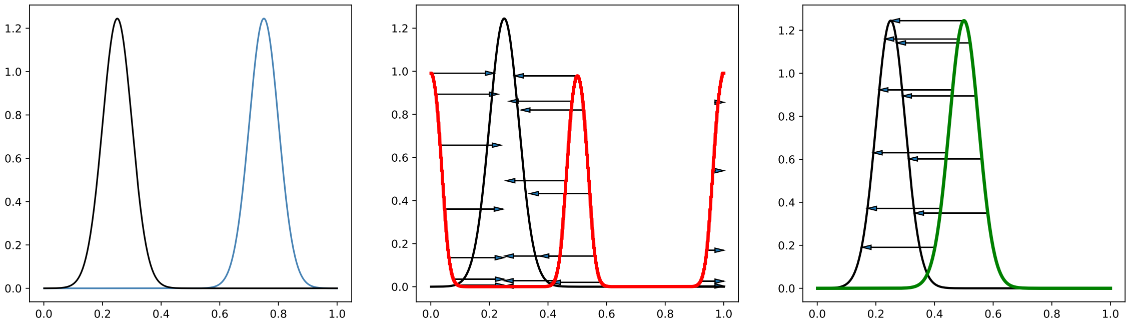


Figure 12.5.1: From left to right: two measures on $\mathbb{S}_1$ (unrolled on $[0, 1]$), the barycenter on $\mathbb{S}_1$ (red) and its transport to the leftmost measure, the barycenter on $\mathbb{R}$ and its transport to the leftmost measure

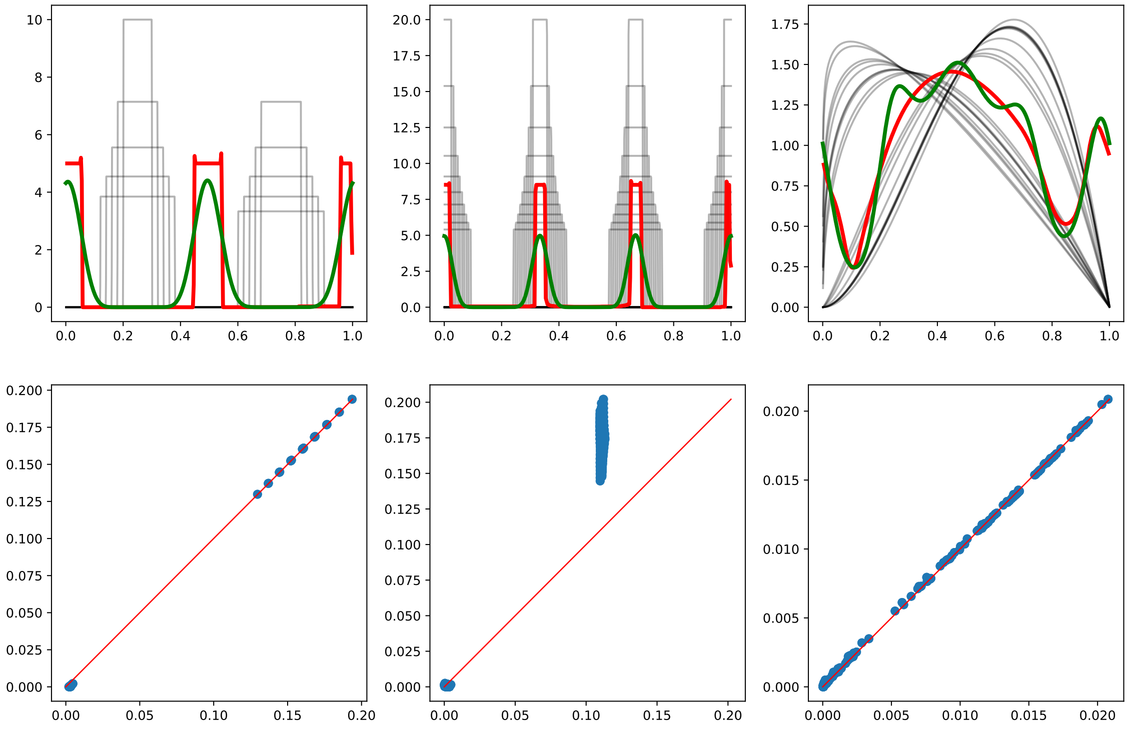


Figure 12.5.2: Top row: densities of the $\widetilde{\mu}_j$'s on $[0, 1]$, and of the Wasserstein and Sinkhorn barycenters (red and green line respectively). Bottom row: Wasserstein distance vs $d_{\log}$ for every possible couple of measures.

this specific example, the cost (on $\mathbb{S}_1$) of transporting the "correct" barycenter on the two measures is 30% lower than the cost of transporting the "Euclidean" one.

We now give some examples of barycenters. In what follows, we use $\bar{\mu}$ to represent the measure on $\mathbb{S}_1$ returned from Algorithm 3 and $\widetilde{\bar{\mu}}$ the associated periodic measure on $\mathbb{R}$. In some cases, it is intuitive what should be the barycenter and we show that our algorithm correctly converges to it. In other ones, intuition fails but we still might get an idea of the goodness of the approximation of the barycenter by comparing the Wasserstein distances $W_2(\mu_i, \mu_j)$ and $d_{\log}(\mu_i, \mu_j) := \|\widetilde{T}_i - \widetilde{T}_j\|_{L_2(\widetilde{\bar{\mu}}_{|[0,1]})}$. Intuitively if $W_2(\mu_i, \mu_j) \approx d_{\log}(\mu_i, \mu_j)$, the tangent plane at $\widetilde{\bar{\mu}}$ has (very) low curvature, so that the problem of finding the Wasserstein barycenter reduces to averaging the quantiles. Therefore, the output of Algorithm 3 should be accurate. Moreover, we also compare the output of Algorithm 3 with the so-called Sinkhorn barycenter (Cuturi and Doucet, 2014; Janati et al., 2020) as implemented in the `Python` package `ott-jax` (Cuturi et al., 2022). To compute the Sinkhorn barycenter, we approximate each measure with an atomic measure with $1,000$ equispaced support points on $[0, 1)$ giving to each point x_i a weight proportional to $\mu(\mathrm{d}x_i)$. Informally, we should expect the Wasserstein and Sinkhorn barycenters to be similar, but the Sinkhorn barycenter should be smoother due to the regularization term involved in the Sinkhorn divergence.

We consider three simulated datasets as follows. Let $\mathcal{U}(c, w)$ denote the uniform measure centered in c and with width w, i.e. the uniform measure over $(c - w/2, c + w/2)$. In the first example, the measures are

$$\begin{aligned}
\widetilde{\mu}_i &= \mathcal{U}\left(0.25, 0.1 + 0.05i\right), \quad i = 1, \ldots, 5 \\
\widetilde{\mu}_i &= \mathcal{U}\left(0.75, 0.1 + 0.05(i-5)\right), \quad i = 5, \ldots, 10
\end{aligned}$$

and extended periodically over the whole $\mathbb{R}$. In the second one instead

$$\begin{aligned}
\widetilde{\mu}_i &= \mathcal{U}\left(0, 0.05 + 0.015i\right), \quad i = 1, \ldots, 10 \\
\widetilde{\mu}_i &= \mathcal{U}\left(1/3, 0.05 + 0.015(i-10)\right), \quad i = 11, \ldots, 20 \\
\widetilde{\mu}_i &= \mathcal{U}\left(2/3, 0.05 + 0.015(i-20)\right), \quad i = 21, \ldots, 30
\end{aligned}$$

In the third case instead, we generate the $\widetilde{\mu}_i$'s by first considering Beta distributions on $(0, 1)$ with parameters $(a_i, 2)$ and then taking their periodic extension. Specifically, $a_i \sim \mathcal{U}(1.3, 0.2)$ for $i = 1, \ldots, 10$ and $a_i \sim \mathcal{U}(2.6, 0.4)$ for $i = 11, \ldots, 20$. Figure 12.5.2 reports the Wasserstein barycenters as found by Algorithm 3 and the Sinkhorn ones for three different simulated datasets. We can see that the Wasserestein ans Sinkhorn barycenters agree and that the Sinkhorn ones are generally smoother. Moreover, in the first and third example the log and Wasserstein distances are indistinguishable which suggests the convergence of Algorithm 3, while in the second example there are some discrepancies.

12.5.2 Eye Dataset

We now present a preliminary investigation on the OCT measurements of NRR thickness. We report a graphical illustration of the measures in the dataset in Figure 12.1.1. Moreover, we also show the barycenter as computed by Algorithm 3. Looking at the Wasserstein and L_2 distances in the tangent plane, we see that these quantities agree for almost all the couples of datapoints, thereby validating the use of the red measure in Figure 12.1.1 as centering point for our PCA. The first two principal directions are reported in Figure 12.5.4. We see that these decouple the variability along the x and y axes.

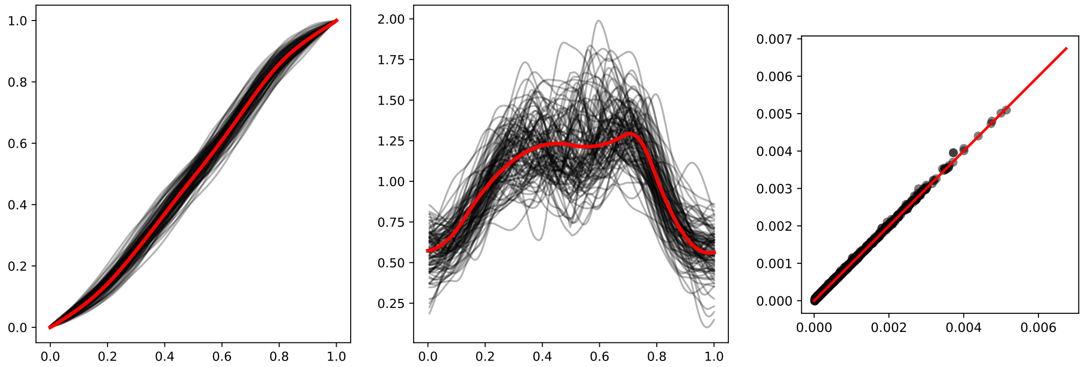


Figure 12.5.3: From left to right: cdfs of the eye's dataset measures (red line denotes the barycenter), pdfs of the eye's dataset measures (red line denotes the barycenter), Wasserstein distance against $d_{\log}$ in the tangent space at the barycenter.

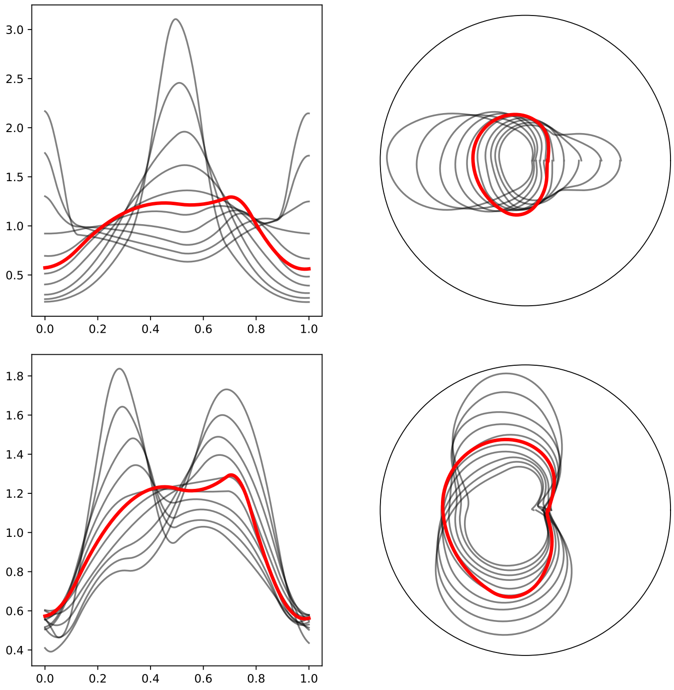


Figure 12.5.4: First (top row) and second (bottom row) principal directions: we report the pdfs on $[0, 1]$ (left panels) and in a polar plot (right panel). The red line denotes the barycenter.

12.6 Discussion

In this chapter we tackled the problem of analyzing distributional data supported on the circumference. Following recent trends in statistics and machine learning, we set out to use the Wasserstein distance to compare probability distributions. To this end, we studied the optimal transportation problem on $\mathbb{S}_1$ and established several new theoretical results, which could also be of independent interest. In particular, we provide an explicit characterization of the optimal transport maps. This result is rather surprising given that optimal transport on Riemannian manifolds us not well established and that the only case where such explicit formulas exist is for measures on the real line. We further explored the weak Riemannian structure of the Wasserstein space and established strong continuity results for the exponential and logarithmic maps, as well as an explicit characterization of the image of the logarithmic map.

Building on our theoretical findings, we proposed a counterpart of the convex PCA in Bigot et al. (2017) for measures on $\mathbb{S}_1$. Following the approach in Pegoraro and Beraha (2022), we propose a numerical method to compute the principal directions by means of a B-spline expansion, which leads to an easily implementable numerical algorithm.

Our definition of PCA requires a "central point", which is usually set equal to the barycenter. We used the algorithm in Zemel and Panaretos (2019) to approximate the Wasserestein barycenter. However, we have not been able to prove the convergence of this algorithm in our setting. Despite numerical simulations do seem to validate the use of Algorithm 3, the theoretical analysis is still an open problem.

Appendix

12.A Technical Preliminaries

12.A.1 Measure Theoretic Preliminaries

Let (M, g) be a Riemannian manifold of dimension n, with TM being its tangent bundle and TM^* its cotangent bundle. We know by definition that g is a section $g : M \to (TM \otimes TM)^*$ and the volume form $\omega : M \to \wedge^n (TM)^*$ is defined locally by $\omega = |\det(g)|^{1/2} dx_1 \wedge \ldots \wedge x_n$.

Let $\mathcal{L}$ be the Lebesgue measure on $\mathbb{R}^n$, we consider the σ-algebra generate by all sets A such that $\varphi(A \cup U)$ is in the Lebesgue σ-algebra of $\mathbb{R}^n$ for some chart (U, φ). Then we indicate with $\mathcal{L}_M$ Riemann-Lebesgue volume measure, i.e. the measure on M such that for every chart (U, φ) and $A \subset U$ contained in the σ-algebra just define:

$$\mathcal{L}_M(A) = \int_{\varphi(A)} |\det(g(\varphi^{-1}))|^{1/2} d\mathcal{L} \tag{12.16}$$

Note that, in general, $\varphi \# \mathcal{L}_M \neq \mathcal{L}$.

Consider $h : M \to \mathbb{R}$ such that $\operatorname{supp}(h) \subset U$, with (U, φ) being a chart, we can integrate h as follows:

$$\int_M h d\mu = \int_U h d\mu = \int_U h f_\mu d\mathcal{L}_M = \int_{z(U)} |\det(g(\varphi^{-1}))|^{1/2} h(\varphi^{-1}) f_\mu(z^{-1}) d\mathcal{L}. \tag{12.17}$$

The general case is defined in a natural way through a partition of unity.

Now we can consider a measure μ on M, with density function f_μ wrt $\mathcal{L}_M$, that is:

$$\mu(A) = \int_A f_\mu d(\mathcal{L}_m) = \int_{\varphi(A)} |\det(g(\varphi^{-1}))|^{1/2} f_\mu(\varphi^{-1}) d\mathcal{L}. \tag{12.18}$$

Lastly, if μ doesn't have a density function wrt $\mathcal{L}_M$, to integrate some function against μ we pick a weak converging sequence $\mu_n \rightharpoonup \mu$ such that μ_n has a density function and extend the definition taking the limit of the integrals.

12.A.2 McCan's Result

Let us recall the definition of c-concavity. Let $c : M \times M \to \mathbb{R} \cup +\infty$. For a function $\psi : M \to \mathbb{R} \cup \{-\infty\}$ define its c-transform $\psi^{c_+} : M \to \mathbb{R} \cup \{-\infty\}$ as

$$\psi^{c_+}(x) = \inf_{y \in M} c(x, y) - \psi(y).$$

Note that this generalizes the Legendre transform, which is recovered when $M = \mathbb{R}^d$ and $c(x, y) = \langle x, y \rangle$.

Definition 2. *A function $\phi : M \to \mathbb{R} \cup \{-\infty\}$ is c-concave if its not identically $-\infty$ and there exists $\psi : M \to \mathbb{R} \cup \{-\infty\}$ such that*

$$\phi = \psi^{c_+}$$

Given $\mu \in \mathcal{W}_2(M)$ and $U \subset M$ open we define $S(U) = \{v : U \to TM \mid \pi \circ v = \mathrm{Id}_U\}$ be the sheaf of local sections of the tangent bundle of M, that is the vector space of tangent vector fields on U. Whenever U is a local trivialization of the tangent bundle, we may use the notation $v_z := v(z) \in T_zM$ for $v \in S(U)$. Now we can define the following sheaf of functions:

$$L^2_\mu(U) = \{v \in S(U) \mid \int \|v_z\|^2 d\mu(z) < \infty\}, \tag{12.19}$$

where $\|v_z\|^2$ stands for $g(v_z, v_z)$.

For any $v \in L^2_\mu(U)$ we can consider the map $\exp(v)$ defined as $\exp(v)(z) := \exp_z(v_z)$, $z \in U$. McCann (2001) proved that if μ is absolutely continuous with respect to the volume measure on M, the unique optimal plan γ^o between μ and ν is induced by a map, i.e. we have $T : M \to M$, inducing $(\mathrm{Id}, T) : M \to M \times M$, such that $\gamma^o = (\mathrm{Id}, T)\#\nu$. Moreover, the map T has the form $T = \exp(-\nabla\phi)$ where ϕ is a d^2-concave function (Gigli, 2011).

12.B Proofs

Let us define

$$d_{\mathbb{Z}}(x,y)^2 := \inf_{p \in Z}(x - y - p)^2 \leq (x-y)^2$$

Note that $d_R(z, z') = d_{\mathbb{Z}}(\log_c(z), \log_c(z'))$

12.B.1 Proof of Theorem 12.2

Proof. The proof follows from the notion of locally optimal plans in Delon et al. (2010). Let γ_θ be the transport plan that takes an element of mass from position $F^-_{\widetilde{\nu}}(u)$ to position $(F^\theta_{\widetilde{\mu}})^-(u)$. Then γ_θ is locally optimal and the associated cost is

$$C_{[\mu,\nu]}(\theta) = \int_0^1 \Big(F^-_{\widetilde{\mu}}(u) - (F^\theta_{\widetilde{\nu}})^-(u)\Big)^2 \mathrm{d}u$$

The (global) optimal plan is associated to $\theta^* = \arg\min C(\theta) = W_2^2(\mu,\nu)$. To recover the optimal transport map we operate the change of variables $x = F^-_{\widetilde{\mu}}(u)$, which yields:

$$W_2^2(\mu,\nu) = \int_0^1 \Big(T^{\widetilde{\nu}}_{\widetilde{\mu}}(x) - x\Big)^2 \mathrm{d}\widetilde{\mu}(x) \geq \tag{12.20}$$

$$\int_0^1 d^2_{\mathbb{Z}}(T^{\widetilde{\nu}}_{\widetilde{\mu}}(x), x)\mathrm{d}\widetilde{\mu}(x) = \tag{12.21}$$

$$\int_{\mathbb{S}_1} d^2_R(\exp_c(T^{\widetilde{\nu}}_{\widetilde{\mu}}(\log_c(z))), z)^2 \mathrm{d}\mu(z) \geq W_2^2(\mu,\nu) \tag{12.22}$$

where the first equality follows by defining $T^{\widetilde{\nu}}_{\widetilde{\mu}} := (F^{\theta^*}_{\widetilde{\nu}})^- \circ F_{\widetilde{\mu}}$, while the last equality is obtained with $z = \exp_c(x)$ and the properties of $d_{\mathbb{Z}}$. □

12.B.2 Proof of Theorem 12.3

Proof. First we observe that:

$$(F^{\theta^*}_{\widetilde{\nu}})^- \big(F_{\widetilde{\mu}}(u+p)\big) = (F^{\theta^*}_{\widetilde{\nu}})^- \big(F_{\widetilde{\mu}}(u) + p\big) = (F_{\widetilde{\nu}})^- \big(F_{\widetilde{\mu}}(u) + p - \theta^*\big) = (F^{\theta^*}_{\widetilde{\nu}})^- \big(F_{\widetilde{\mu}}(u)\big) + p$$

which means that $\widetilde{T}(u+p) = \widetilde{T}(u) + p$ for every integer p.

By Theorem 12.2 we know that $\widetilde{T}$ is an optimal transport map if and only if $\theta = \theta^*$ as in Equation (12.8). Define:

$$C_{[\mu,\nu]}(\theta) = \int_0^1 \left(F_{\widetilde{\mu}}^{-}(u) - (F_{\widetilde{\nu}}^{\theta})^{-}(u) \right)^2 \mathrm{d}u$$

Delon et al. (2010) prove that the map $\theta \mapsto C_{[\mu,\nu]}(\theta)$ is strictly convex if μ is a.c.. Thus θ^* is the unique stationary point of the function. For this reason we compute the derivative of $C_{[\mu,\nu]}$ in θ, knowing that θ^* is the only value such that $(C_{[\mu,\nu]})' = 0$. Thanks to Leibniz rule we can write:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\theta} C_{[\mu,\nu]}(\theta) &= \int_0^1 \frac{\mathrm{d}}{\mathrm{d}\theta} \left(F_{\widetilde{\mu}}^{-}(u) - (F_{\widetilde{\nu}})^{-}(u-\theta) \right)^2 \mathrm{d}u \\
&= \int_0^1 \frac{\mathrm{d}}{\mathrm{d}\theta} \left(F_{\widetilde{\mu}}^{-}(u+\theta) - (F_{\widetilde{\nu}})^{-}(u) \right)^2 \mathrm{d}u \\
&= \int_0^1 2 \left(F_{\widetilde{\mu}}^{-}(u+\theta) - (F_{\widetilde{\nu}})^{-}(u) \right) \frac{1}{f_{\widetilde{\mu}}(F_{\widetilde{\mu}}^{-}(u+\theta))} \mathrm{d}u
\end{aligned}$$

with the change of variables $v = F_{\widetilde{\mu}}^{-}(u+\theta)$, which entails $F_{\widetilde{\mu}}(v) - \theta = u$ and $\mathrm{d}u = \mathrm{d}\widetilde{\mu}(v) = f_{\widetilde{\mu}}(v)\mathrm{d}v$ we obtain

$$= \int_{v_0}^{1+v_0} 2 \left(v - \widetilde{T}(v) \right) \frac{f_{\widetilde{\mu}}(v)}{f_{\widetilde{\mu}}(v)} \mathrm{d}v$$

with $v_0 = F_{\widetilde{\mu}}^{-}(\theta)$. Optimality follows if and only if such quantity is equal to zero and thus:

$$0 = \int_{v_0}^{1+v_0} \left(\widetilde{T}(v) - v \right) \mathrm{d}v = \int_{v_0}^{1} \left(\widetilde{T}(v) - v \right) \mathrm{d}v + \int_{1}^{1+v_0} \left(\widetilde{T}(v) - v \right) \mathrm{d}v$$

Via the change of variables $u = v - 1$ we have:

$$-2 \int_0^{v_0} \left(\widetilde{T}(u) - 1 - (u-1) \right) \mathrm{d}v$$

and thus:

$$-2 \int_0^{1} \left(\widetilde{T}(u) - u \right) \mathrm{d}u$$

□

12.B.3 Proof of Theorem 12.4

Proof. Observe that T is an optimal transport map, then defining

$$\widetilde{T}(x) := \log_c \circ T \circ \exp_c,\ x \in (0,1), \qquad \widetilde{T}(x+p) = \widetilde{T}(x) + p,\ p \in \mathbb{Z}$$

clearly satisfies the monotonicity and "periodicity" requirements. Moreover, (12.10) is satisfied by Theorem 12.3. To prove that $|\widetilde{T}(x) - x| < 1/2$ note that this is equivalent to $\inf_{p \in \mathbb{Z}} |\,\widetilde{T}(x) - x - p\,| = |\,\widetilde{T}(x) - x\,|$. Since $T : \mathbb{S}_1 \to \mathbb{S}_1$ is an optimal transport map from μ to ν we have:

$$\begin{aligned}
W_2(\mu,\nu) &= \int_{\mathbb{S}_1} d_R(T(z), z)^2 d\mu \\
&= \int_{[0,1]} d_R(T(\exp_c(x)), \exp_c(x))^2 d(\log_c \# \mu)(x) \\
&= \int_{[0,1]} d_{\mathbb{Z}}(\widetilde{T}(x), x)^2 d\widetilde{\mu}(x)
\end{aligned}$$

where the first equality is obtained via the definition of optimal transport map, the second through the change of variables $z = \exp_{c\,|\,[0,1]}(x)$, and in the last one we use the definition of $\widetilde{T}$, $\widetilde{\mu}$ and the properties of $d_{\mathbb{Z}}$. As already noted, we have $\inf_{p\in\mathbb{Z}} |\,\widetilde{T}(x) - x - p\,| \leq |\,\widetilde{T}(x) - x\,|$. If the strict inequality holds for some $A \subset [0,1]$ with $\widetilde{\mu}(A) > 0$ then also the integrals on $[0,1]$ must be different, and the thesis follows.

To prove the reverse statement, it suffices to prove that $\widetilde{T}(v)$ can be written as $F^{-}_{\widetilde{\nu}}(F_{\widetilde{\mu}}(v)+\theta)$, which is equivalent to saying that

$$F^{-}_{\widetilde{\nu}}(v) = \widetilde{T}\left((F_{\widetilde{\mu}}(\cdot)+\theta)^{-1}(v)\right) = \widetilde{T}(F^{-}_{\widetilde{\mu}}(v-\theta)).$$

Define $G_{\widetilde{\nu}} := \widetilde{T} \circ F^{-}_{\widetilde{\mu}}$, then of course $\widetilde{T} = G_{\widetilde{\nu}} \circ F_{\widetilde{\mu}}$. We show that $G_{\widetilde{\nu}}(u) \equiv F^{-}_{\widetilde{\nu}}(u+\theta)$. We have that, for $x \in [0,1)$

$$F_{\widetilde{\nu}}(x) = \widetilde{\nu}([0,x]) = \widetilde{\mu}(\widetilde{T}^{-1}([0,x])) = \widetilde{\mu}([\widetilde{T}^{-1}(0), \widetilde{T}^{-1}(x)]) = F_{\widetilde{\mu}}(\widetilde{T}^{-1}(x)) - F_{\widetilde{\mu}}(\widetilde{T}^{-1}(0))$$

and observe that $F_{\widetilde{\nu}}(x) \leq 1$ thanks to $|\widetilde{T}(x) - x| < 1/2$. Hence, the pushforward of $\widetilde{\mu}$ on $\mathbb{S}_1$ gives a valid probability measure. Taking the inverse of $F_{\widetilde{\nu}}$

$$F^{-}_{\widetilde{\nu}}(u) = \left(F_{\widetilde{\mu}}(\widetilde{T}^{-1}(\cdot)) - F_{\widetilde{\mu}}(\widetilde{T}^{-1}(0))\right)^{-}(u) = \widetilde{T} \circ F^{-}_{\widetilde{\mu}}(u + F_{\widetilde{\mu}}(\widetilde{T}^{-1}(0)))$$

and setting $-\theta = F_{\widetilde{\mu}}(\widetilde{T}^{-1}(0))$ yields the result. □

12.B.4 Proof of Theorem 12.5

To prove item (ii), we will need the two following preliminary lemmas.

Lemma 12.1. *Suppose we have $W_2(\nu, \nu_n) \to 0$ in $\mathcal{W}_2(\mathbb{S}_1)$ with ν, ν_n being a.c. wrt $\mathcal{L}_{\mathbb{S}_1}$ (for every n). Then $W_2(\log_c \#\nu, \log_c \#\nu_n) \to 0$ in $\mathcal{W}_2(\mathbb{R})$.*

Proof. From Theorem 7.12 in Villani (2003), convergence in the Wasserstein metric is equivalent to weak convergence plus the tightness condition: there exist x_0 such that

$$\lim_{R\to+\infty} \limsup_{k\to+\infty} \int_{d(x,x_0)>R} d(x,x_0)^p d \log_c \#\nu_k(x)$$

Observe that each measure $\log_c \#\nu_k$ is supported on $[0,1]$ so that the condition is always met. Hence, we just need to show that the sequence $\log_c \#\nu_k$ converges weakly. For measures on the real line, weak convergence is equivalent of pointwise convergence of the associated distribution functions at continuity points. That is, letting $F_k(x) := \log_c \#\nu_k([0,x))$ and $F(x) := \log_c \#\nu([0,x))$, it must hold that

$$F_k(x) \to F(x), \qquad \text{all } x \text{ such that } F(x) \text{ is continuous} \tag{12.23}$$

Observe that $F_k(x) = \nu_k(\exp_c([0,x)))$ by definition. By Portmanteau's theorem, for any x such that $\nu(\{\exp_c(x)\}) = 0$ we have that $\nu_k(\exp_c([0,x))) \to \nu_k(\exp_c([0,x)))$ which easily implies (12.23)

□

Lemma 12.2. *Suppose we have $W_2(\nu, \nu_n) \to 0$ with μ, ν, ν_n being a.c. wrt $\mathcal{L}_{\mathbb{S}_1}$ (for every n). Then $\|\log_\mu(\nu_n) - \log_\mu(\nu)\|_{L^2_\mu} \to 0$.*

Proof. By Lemma 12.1 we have $F_{\widetilde{\nu}_n}(x) \to F_{\widetilde{\nu}}(x)$ and the same for the quantile functions. As a consequence $C_{[\nu_n,\mu]}(\theta) \to C_{[\nu,\mu]}(\theta)$.

Thus consider $\theta_n = \arg\min_K C_{[\nu_n,\mu]}$. Since $\{\theta_n\} \subset K$ compact, we can consider a converging subsequence which we still call $\{\theta_n\}$ with an abuse of notation. Let $\theta_n \to \theta^*$. As shown in Delon et al. (2010) $C_{[\nu_n,\mu]}$ is strictly convex. Thus, by standard arguments, we conclude that $\theta^* = \arg\min C_{[\nu,\mu]}$.

Now consider:

$$\mid \Big(F_{\widetilde{\mu}}(F^{-}_{\widetilde{\nu}_n}(u+\theta_n))\Big) - \big(F_{\widetilde{\mu}}(F^{-}_{\widetilde{\nu}}(u+\theta^*))\big) \mid \leq \tag{12.24}$$

$$\mid \Big(F_{\widetilde{\mu}}(F^{-}_{\widetilde{\nu}_n}(u+\theta_n))\Big) - \big(F_{\widetilde{\mu}}(F^{-}_{\widetilde{\nu}}(u+\theta_n))\big) \mid + \mid \big(F_{\widetilde{\mu}}(F^{-}_{\widetilde{\nu}}(u+\theta_n))\big) - \big(F_{\widetilde{\mu}}(F^{-}_{\widetilde{\nu}}(u+\theta^*))\big) \mid \tag{12.25}$$

Which implies the pointwise convergence $T^{\widetilde{\mu}}_{\widetilde{\nu}_n}(u) \to T^{\widetilde{\mu}}_{\widetilde{\nu}}(u)$: both addends in the last sum go to 0. Since these maps are continuous and bounded on $[0,1]$ we have uniform convergence and strong convergence. The strong convergence in the image of $\log_\mu$ then follows. □

We are now ready to prove Theorem 12.5

Proof.

- To check the continuity of $\exp_\mu$, consider $T^{\nu_1}_\mu \times T^{\nu_2}_\mu : \mathbb{S}_1 \to \mathbb{S}_1 \times \mathbb{S}_1$ and induce the transport plan $\gamma = (T^{\nu_1}_\mu, T^{\nu_2}_\mu)\#\mu$. Then we have:

$$\begin{aligned} W_2^2(\nu_1,\nu_2) &\leq \int_{\mathbb{S}_1\times\mathbb{S}_1} d_R(z,w)^2 d\gamma(dzdw) \\ &= \int_{\mathbb{S}_1} d_R(T^{\nu_1}_\mu(z), T^{\nu_2}_\mu(z))^2 d\mu(dz) \\ &\leq \int_{[0,1]} d_{\mathbb{Z}}(T^{\widetilde{\nu}_1}_{\widetilde{\mu}}(x), T^{\widetilde{\nu}_2}_{\widetilde{\mu}}(x))^2 d\widetilde{\mu}(dx) \\ &\leq \parallel T^{\widetilde{\nu}_1}_{\widetilde{\mu}} - T^{\widetilde{\nu}_2}_{\widetilde{\mu}} \parallel^2_{L^2_{\widetilde{\mu}}([0,1])} = \parallel \log_\mu(\nu_1) - log_\mu(\nu_2) \parallel^2_{L^2_\mu} . \end{aligned}$$

  where the last identity is obtained thanks to $\log_c \#\mu = \widetilde{\mu}$ on $[0,1]$.

- To check the continuity of $\log_\mu$ instead,

  By an approximation argument we obtain sequential continuity of $\log_\mu$ at any measure $\nu \in \mathcal{W}_2(\mathbb{S}_1)$: consider $\nu_n \to \nu$, with ν_n a.c. measures. Then $\{\log_\mu(\nu_n)\}$ is a Cauchy sequence in L^2_μ, which is a complete metric space, and so it converges to a vector field v. Consider $\exp_\mu(v)$. By the continuity of $\exp_\mu$ we have $\nu_n \to \exp_\mu(v)$ which then entails $\exp_\mu(v) = \nu$.

  Lastly, sequential continuity in metric spaces implies continuity.

□

13. BayesMix: Bayesian Mixture Models in C++

We describe **BayesMix**, a C++ library for MCMC posterior simulation for general Bayesian mixture models. The goal of **BayesMix** is to provide a self-contained ecosystem to perform inference for mixture models to computer scientists, statisticians and practitioners. The key idea of this library is *extensibility*, as we wish the users to easily adapt our software to their specific Bayesian mixture models. In addition to the several models and MCMC algorithms for posterior inference included in the library, new users with little familiarity on mixture models and the related MCMC algorithms can extend our library with minimal coding effort. Our library is computationally very efficient when compared to competitor software. Examples show that the typical code runtimes are from two to 25 times faster than competitors for data dimension from one to ten. Our library is publicly available on Github at https://github.com/bayesmix-dev/bayesmix/.

13.1 Introduction

Mixture models are a popular framework in Bayesian inference, being particularly useful for density estimation and cluster detection; see Fruhwirth-Schnatter et al. (2019) for a recent review. Mixture models are convenient as they allow to decompose complex data-generating processes into simpler pieces, for which inference is easier. Moreover, they are able to capture heterogeneity and to group data together into homogeneous clusters. The usefulness of mixture models, either finite or infinite, is evident from the huge literature developed around this topic, with applications in genomics (Elliott et al., 2019), healthcare (Beraha et al., 2022), text mining (Blei et al., 2003) and image analysis (Lü et al., 2020), to cite a few. See also Mitra and Müller (2015) for Bayesian nonparametric mixture models in biostatistical applications and the last five chapters in Fruhwirth-Schnatter et al. (2019) for applications of mixture models to different contexts, including industry, finance, and astronomy.

In a mixture model, each observation is assumed to be generated from one of m groups or populations, with m finite or infinite, and each group suitably modelled by a density, typically from a parametric family. We consider data $y_1, \dots, y_n \in \mathbb{Y} \subset \mathbb{R}^d$, $d \geq 1$. To define a mixture model we take weights $\boldsymbol{w} = (w_1, \dots, w_m)$ such that $w_h \geq 0$ for all $h = 1, \dots, m$, $\sum_h w_h = 1$, component-specific parameters $\boldsymbol{\tau} = (\tau_1, \dots, \tau_m) \in \Theta^m$, with $m < +\infty$ or $m = +\infty$, and a parametric kernel $f(\cdot \mid \cdot)$ such that $f(\cdot \mid \tau)$ is a density on $\mathbb{Y}$ for each τ in Θ Specifically, we assume

$$y_i \mid \boldsymbol{w}, \boldsymbol{\tau} \overset{\text{iid}}{\sim} p(y) := \sum_{h=1}^{m} w_h f(y \mid \tau_h), \qquad i = 1, \dots, n \ . \tag{13.1}$$

In this chapter we consider mixture models under the Bayesian approach, so that the model is completed with a prior for $(\boldsymbol{w}, \boldsymbol{\tau})$ and m, i.e.

$$\boldsymbol{w}, \boldsymbol{\tau}, m \sim \pi(\boldsymbol{w}, \boldsymbol{\tau}, m) \ . \tag{13.2}$$

Posterior simulation for $(\boldsymbol{w}, \boldsymbol{\tau}, m)$ under model (13.1)-(13.2) is extremely challenging. First of all, the posterior is multimodal due to the well-known label switching problem.

Second, the number of parameters is typically huge and possibly infinite. Several Markov chain Monte Carlo algorithms, specific for Bayesian mixture models, have been proposed since the early 2000s for posterior simulation, as, e.g., Neal (2000) and Ishwaran and James (2001). Nonetheless, as we discuss more in detail in Section 13.2, only a handful of packages are available to practitioners nowadays as, for instance, the recent **BNPmix** R package (Corradin et al., 2020) and the popular **DPpackage** (Jara et al., 2011). This type of packages often provides either an R or a Python interface to some C++ code, hence being usually efficient in fitting the associated model.

Given the generality of (13.1)-(13.2), it is unrealistic to expect that a single package can be used to fit *any* mixture model. In particular, the choice of the parametric kernel $f(\cdot \mid \cdot)$ is prescribed by the type of data (e.g. unidimensional vs multidimensional, continuous, categorical, counts) of the study. Many packages are built only for some type of data, and hence some kernels and priors, so that, it is likely that statisticians need to consider different models from the ones already available in potentially interesting software packages. In addition, the C++ core code is usually not written in order to be extended, with poor documentation, thus resulting in a code that is hard to make use for extensions.

To overcome these limitations, we describe here **BayesMix**, a C++ library for Markov chain Monte Carlo (MCMC) simulation in Bayesian nonparametric (BNP) mixture models. The ultimate goal of **BayesMix** is to provide statisticians a self-contained ecosystem to perform inference for mixture models. In particular, the driving idea behind this library is *extensibility*, as we wish statisticians to easily adapt our software to their needs. For instance, changing the parametric kernel f in (13.1) can be accomplished by defining a class specific to that kernel, which usually requires less than 30 lines of C++ code. This new class can be seamlessly integrated in the **BayesMix** library and, used in combination with prior distributions for the rest of the parameters and algorithms for posterior inference which are already present. Similarly, defining a new prior for $\boldsymbol{w}$ requires only to implement a class for that prior, and so on. Therefore, new users with little familiarity on mixture models and the related MCMC algorithms can easily extend our library with minimal coding effort.

The extensibility of **BayesMix** does not come with a compromise on the efficiency. For instance, compared to **BNPmix** package, when running the same MCMC algorithm, our code runtimes are typically two times faster when y_i is univariate and approximately 25 times faster when y_i is four-dimensional. Typical indicators of the efficiency of MCMC algorithms such as autocorrelation and effective sample size confirm that the performance obtained with our library is superior not only from the runtime point of view, but also in terms of the overall quality of the MCMC samples. Moreover, we show that our implementation is able to scale to moderate and high dimensional settings and that **BNPmix** fails to recover the underlying signal when y_i is ten-dimensional, unlike our library.

As far as software is concerned, we achieve the desired customizability, modularity and extensibility through an object-oriented approach, making extensive use of static and runtime polymorphism through class templates and inheritance. This may constitute a barrier for new users wishing to extend our library, as knowledge of those C++ programming techniques is undoubtedly required. In Section 13.7 we give an example on how to implement a completely new mixture model in the library, which requires less than 130 lines of code. Then, new users can exploit this example and adapt it to their needs.

We point out that at this stage, **BayesMix** is not R package, but a very powerful and flexible C++ library. Although we provide a Python interface (see Section 13.5), this is simply a wrapper around the C++ executable. A more sophisticated Python package is currently under development and available at https://github.com/bayesmix-dev/pybmix, but its description is beyond the scope of this chapter.

The rest of this article is organized as follows. Section 13.2 reviews software to fit

Bayesian mixture models. Section 13.3 gives background on two of the algorithms we have included in the library, to better understand the description of the different modules of the **BayesMix** library in Section 13.4. Section 13.5 shows how to install and use the library by examples. Benchmark datasets are fitted to our library and the competitor R package **BNPmix** in Section 13.6. Section 13.7 contains material for more advanced users, i.e., we show how new developers could extend the library. The article concludes with a discussion in Section 13.8.

13.2 Review of available software

One of the main drawbacks of Bayesian inference is that MCMC methods can be extremely demanding from the computational point of view. Moreover, the design of efficient MCMC algorithms and their practical implementation is not a trivial task, and thus might preclude the use of these methods to non-specialists. Nonetheless, Bayesian statistics has greatly increased in popularity in recent years, thanks to the growth of computational power of computers and the development of several dedicated software products.

In this section, we review in particular two packages for Bayesian mixture models, namely the **DPpackage** and the **BNPmix** R packages. They do not exhaust all the possibilities, but they are, among all software, the packages which implement the same models as in **BayesMix** via the same algorithms. Other choices include using probabilistic programming languages such as JAGS (Plummer, 2003) and Stan (Carpenter et al., 2017), though their review is beyond the scope of this chapter. We limit ourselves to note that Stan simulates from the posterior through Hamiltonian Monte Carlo while JAGS uses Gibbs sampling. **BayesMix** uses part of the Stan `math` library for evaluating distributions, random sampling and automatic differentiation. Observe that it is straightforward to compute the posterior of finite mixture models via JAGS or Stan. However, since those probabilistic programming languages work for a large class of Bayesian models, they can be less computationally efficient and fast than software purposely designed for Bayesian mixture models.

In addition to the **DPpackage** and **BNPmix**, other R packages are available to fit mixture models. We report here **BNPdensity** (Arbel et al., 2020; Barrios et al., 2013) and **dirichletprocess** (Ross and Markwick, 2020). The former focuses on nonparametric mixture models based on normalized completely random measures, using the Ferguson-Klass algorithm. The latter focuses on Dirichlet process mixture models. Both the packages are very flexible and implement several models and algorithms. However, they are written entirely in the R language, which comes as a serious drawback as far as performance is concerned. We cite here also **NIMBLE** (de Valpine et al., 2017), which is a hybrid between a probabilistic programming language and an R package, and allows to fit Dirichlet process mixture models.

We also mention the Python **bnpy** package (Hughes and Sudderth, 2014), released in 2017. The package exploits BNP models based on the Dirichlet process and finite variations of it, but forgoes traditional MCMC methods in favor of variational inference techniques such as stochastic and memoized variational inference.

The most complete software that fits BNP models is arguably the R library **DPpackage** (Jara et al., 2011). Its most important design goal is the efficient implementation of some popular model-specific MCMC algorithms. For this reason, it exploits embedded C, C++, and Fortran code for posterior sampling. **DPpackage** boasts a large number of features, including, but not limited to, density estimation through both marginal and conditional algorithms, ROC curve analysis, inference for censored data, binary regression, generalized additive models, and longitudinal and clustered data using generalized linear mixed models. The Bayesian models in **DPpackage** are focused on the Dirichlet Process and its variations, e.g. DP mixtures with normal kernels, Linear Dependent DP (LDDP), Linear Dependent Poisson-Dirichlet (i.e., the Pitman-Yor mixture), weight-dependent DP, and Pólya trees models. Unfortunately, this package was orphaned in 2018 by its authors, and has been archived from the Comprehensive R Archive Network (CRAN) database of R packages in 2019.

BNPmix is a recently published R package for Bayesian nonparametric multivariate inference (Corradin et al., 2020). Its focus is on Pitman-Yor mixtures with Gaussian kernels, thus including the Dirichlet process mixture. This package performs density estimation and clustering through several state-of-the-art MCMC methods, namely marginal sampling, slice sampling, and the recent importance conditional sampling, introduced by the same authors (Canale et al., 2019). It also allows regression with categorical covariates, by using the partially exchangeable Griffiths-Milne dependent Dirichlet process (GM-DDP) model as defined in Lijoi et al. (2014b).

The goal of **BNPmix** is to provide a readily usable set of functions for density estimation and clustering under a number of different BNP Gaussian mixture models, while at the same time being highly customizable in the specification of prior information. It also allows for different hyperpriors for the Gaussian mixture models of interest. The underlying structure of the package is written in C++, using **Armadillo** as the linear algebra library of choice, and it is integrated to R through the packages **Rcpp** and **RcppArmadillo**. Inspecting the source code of **BNPmix**, it is clear that the package lacks in modularity since, for every choice of $f(\cdot|\tau)$ and prior distribution $\pi(\boldsymbol{w}, \boldsymbol{\tau})$, an MCMC algorithm is implemented with little sharing of code. As a consequence, new users aiming at extending the library to other mixture models (for instance, to non-Gaussian kernels) face a tough challenge. Since **BNPmix** is a recent R package and it considers some of the mixtures our **BayesMix** considers as well, we extensively compare the two libraries in Section 13.6. However, the scopes and, probably, the end-users of **BNPmix** are different from those of our library as, in our opinion, **BNPmix** is an R package providing a collection of a sort of black-box (i.e. not extensible) methods for density estimation and clustering. The C++ functions are not documented, therefore making it difficult to extend the library to new models for new users. However, for statisticians or practitioners who only intend to fit the models in **BNPmix** to their data, this R package does a very good job.

Key characteristics of good software for Bayesian mixture models thus include flexibility and the ability of providing efficient implementations of popular models. Flexibility also comes from modularity and extensibility, as they allow re-usability of existing code, as well as combination and implementation of brand-new models and algorithms without re-writing the entire environment from scratch. In programming terms, this often translates into the object-oriented paradigm. These are exactly the features we have aimed at implementing into **BayesMix**.

13.3 Bayesian Mixture Models

Throughout this chapter, we consider Bayesian mixture models as in (13.1)-(13.2). For inferential purposes, it is often useful to introduce a set of latent variables $\boldsymbol{c} = (c_1, \dots, c_n)$, $c_i \in \{1, \dots, m\}$ and rewrite (13.1) as:

$$\begin{aligned} y_i \,|\, \boldsymbol{c}, \boldsymbol{\tau} &\overset{\text{ind}}{\sim} f(\cdot \,|\, \tau_{c_i}), \qquad i = 1, \dots, n \\ c_i \,|\, \boldsymbol{w} &\overset{\text{iid}}{\sim} \text{Categorical}(\{1, \dots, m\}, \boldsymbol{w}), \quad i = 1, \dots, n \end{aligned} \tag{13.3}$$

The c_i's are usually referred to as cluster allocation variables, and the clustering of the observations is the partition of $\{1, \dots, n\}$ induced by the c_i's into mutually disjoint sets $C_j = \{i : c_i = h\}$. We refer to m as the number of *components* in the model, and to the cardinality of the set $\{C_j\}_j$ such that C_j is non-empty as the number of *clusters.* Note that the number of clusters might be strictly less then the number of components.

In the Bayesian framework, the likelihood is complemented with prior (13.2) on parameters $\boldsymbol{w}, \boldsymbol{\tau}$ and possibly m. In particular, we distinguish three cases: (i) m is finite and fixed, (ii) m is finite almost surely but random and (iii) $m = +\infty$. Since m can be 'large', these mixtures are considered as belonging to the (Bayesian) nonparametric framework. A popular choice for $f(\cdot \mid \tau)$ is the Gaussian density (unidimensional or multidimensional) with τ given by the mean and the variance (matrix). As an alternative, Student's t, skew-normal, location–scale or gamma densities (in case of positive data points) might be considered. In general, the marginal prior for $\boldsymbol{w}$ is the finite-dimensional Dirichlet distribution when $m < +\infty$ or the stick–breaking distribution when $m = +\infty$. Parameters τ_i's are typically assumed independent and identically distributed (iid) from a suitable distribution. The goal of the analysis is then estimating the posterior distribution of the parameters, i.e., the conditional law of $(\boldsymbol{w}, \boldsymbol{\tau}, m)$ given observations $\boldsymbol{y}$ (when m is fixed we can consider the distribution of m as a degenerate point-mass distribution). Such posterior distribution is not available in closed form and Markov chain Monte Carlo algorithms are commonly employed to sample from it.

Of course, the algorithms for posterior inference will be different depending on the value of m (see above). Case (i) is the easiest, as a careful choice of the marginal priors for $\boldsymbol{w}$ and $\boldsymbol{\tau}$ leads to closed-form expression for the full conditionals, so that inference can be carried out through a simple Gibbs sampler. In case (iii), the whole set of parameters cannot be physically stored in a computer, and algorithms need to rely on marginalization techniques (see, e.g. Neal, 2000; Walker, 2007; Papaspiliopoulos and Roberts, 2008; Kalli et al., 2011; Griffin and Walker, 2011; Canale et al., 2019). Case (ii) requires a transdimensional MCMC sampler (Green, 1995), examples of which are the split-merge reversible jump MCMC (Richardson and Green, 1997) and the birth-death Metropolis-Hastings (Stephens, 2000) algorithm. In the context of our work, we distinguish between *marginal* and *conditional* algorithms. The former marginalize out the $m-k$ non-allocated components from the state space, dealing only with the cluster allocations; examples are the celebrated algorithms by Neal (Neal, 2000). The latter instead store the whole parameters state (or an approximation of it if $m = +\infty$); examples include the Blocked-Gibbs sampler in Ishwaran and James (2001), the retrospective sampler in Papaspiliopoulos and Roberts (2008) and the slice sampler in Walker (2007).

In the remainder of this section, we present two well-known algorithms for posterior inference in detail. This will be useful in Section 13.4 to understand the modules of the **BayesMix** library. For observations $y_1, \ldots, y_n$ we assume the likelihood as in (13.1) (or equivalently as in (13.3)) and assume that $\boldsymbol{w} \sim \pi(\boldsymbol{w})$ and $\tau_h \overset{\text{iid}}{\sim} G_0$, $h = 1, \ldots, m$, where G_0 denotes a distribution over $\Theta \subset \mathbb{R}^p$, for some positive integer p.

13.3.1 A marginal algorithm: Neal's Algorithm 2

Neal (2000) proposes several algorithms for posterior inference for Dirichlet process mixture models. These algorithms have been later extended to work with more general models, such as Normalized Completely Random Measures mixture models (see Favaro and Teh, 2013) and finite mixture models with a random number m of components (see Miller and Harrison, 2018).

The state of the Markov chain consists of $\boldsymbol{c} = (c_1, \ldots, c_n)$ and $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_k)$, k denoting the number of clusters, $k \leq m$. The key mathematical object for this algorithm is the so-called Exchangeable Partition Probability Function (EPPF, Pitman, 1995), that is the prior on the clusters configurations $\{C_1, \ldots, C_k\}$ induced by the prior on the weights $\boldsymbol{w}$, when $\boldsymbol{w}$ is marginalized out. Following Pitman (1995), the probability of realization $C_1, \ldots, C_k$ depends only on their sizes, i.e., $\Phi(n_1, \ldots, n_k)$, where n_h denotes the cardinality of C_h.

Neal's algorithm 2 can be summarized as follows:

1. Sample each cluster allocation variable c_i independently from

$$p(c_i = h \mid \cdots) \propto \begin{cases} \Phi(n_1^{-i}, \dots, n_h^{-i} + 1, \dots n_k^{-i}) f(y_i \mid \tau_h) & \text{for } h = 1, \dots, k \\ \Phi(n_1^{-i}, \dots, n_h^{-i}, \dots n_k^{-i}, 1) m(y_i) & \text{for } h = k + 1 \end{cases}$$

   where n_h^{-i} denotes the cardinality of the h-th cluster when observation i is removed from the state and $m(y_i) = \int_\Theta f(y_i \mid \theta) G_0(\mathrm{d}\theta)$.

2. Sample the cluster-specific values independently from $p(\tau_h \mid \cdots) \propto \prod_{i:c_i=h} f(y_i \mid \tau_h) g_0(\tau_h)$.

Observe that in Step 1., since the $m - k$ non-allocated components and the weights $\boldsymbol{w}$ are integrated out when updating each cluster label c_i, the algorithm either assigns the i-th observation to one of the already existing clusters, or to a new one.

BayesMix allows only for the so-called Gibbs type priors (De Blasi et al., 2013), for which the probability of a new cluster is

$$\Phi(n_1, \dots, n_h, \dots n_k, 1) = f_1(k, n, \theta) \quad \text{and} \quad \Phi(n_1, \dots, n_h + 1, \dots n_k) = f_2(n_h, n, \theta), \quad (13.4)$$

where θ is a (possibly multidimensional) parameter governing the EPPF, n is the total number of observations, and k is the number of clusters. The expression of f_1 and f_2 is specific of each EPPF.

13.3.2 A conditional algorithm: the Blocked Gibbs sampler by Ishwaran and James (2001)

In Neal's Algorithm 2 described in Section 13.3.1 we can assume m to be either finite or infinite, random or fixed, as long as the EPPF is available. For the blocked Gibbs sampler, instead, we need to assume a finite and fixed m.

The state of the algorithm consists of $\boldsymbol{c}, \boldsymbol{w}, \boldsymbol{\tau}$. The algorithm can be summarized as follows:

1. sample the cluster allocations from the discrete distribution over $\{1, \dots, m\}$ such that $p(c_i = h \mid \cdots) \propto w_h f(y_i \mid \tau_h)$ for any i (independently).
2. Sample the weights from $p(\boldsymbol{w} \mid \cdots) \propto \pi(\boldsymbol{w}) \prod_{i=1}^n w_{c_i}$.
3. Sample the cluster-specific parameters independently from

$$p(\tau_h \mid \cdots) \propto G_0(\tau_h) \prod_{i:c_i=h} f(y_i \mid \tau_h), \qquad \text{for any } h.$$

13.4 The **BayesMix** paradigm: extensibility through modularity

In this section, we give a general overview of the main building blocks in **BayesMix**. This is enough for users to understand what is happening behind the curtains. A more detailed explanation of the software, including the class hierarchy and the application programming interfaces (API) for each class can be found in Section 13.7, where we also give a practical example on how to extend the existing code to a new model. The complete documentation of all the functions and classes in our library can be found at https://bayesmix.readthedocs.io.

Let us examine the algorithms in Sections 13.3.1 and 13.3.2. Step 3 in the Blocked Gibbs sampler (Section 13.3.2) and step 2 in Neal's algorithm 2 (Section 13.3.1) are identical. This step depends only on: (*i*) the prior G_0, (*ii*) the likelihood $f(\cdot \mid \cdot)$, and (*iii*) the observations $\{y_i : c_i = h\}$. In the rest of the chapter, by *likelihood* $f(\cdot \mid \cdot)$ we mean the parametric component kernel in (13.1).

The `Hierarchy` module Observe that the update of τ_h is cluster-specific, and it can be performed in parallel over different clusters. This suggests that one of the main building blocks of the code must be able to represent this update. We call these classes `Hierarchies`, since they depend both on the prior g_0 and the likelihood $f(\cdot \mid \cdot)$. In **BayesMix**, each choice of G_0 is implemented in a different `PriorModel` object and each choice of $f(\cdot \mid \cdot)$ in a `Likelihood` object, so that it is straightforward to create a new `Hierarchy` using one of the already implemented priors or likelihoods. The sampling from the full conditional of τ_h is performed in an `Updater` class. When the `Likelihood` and `PriorModel` are conjugate or semi-conjugate, model-specific updaters can be used to sample from the full conditional, either by computing it in closed form or through a Gibbs sampling step. Alternatively, we also provide two off-the-shelf `Updater`s that can be used with any combination of `Likelihood` and `PriorModel`, namely the `RandomWalkUpdater` and the `MalaUpdater`. The former samples from the full conditional of τ_h via a random-walk Metropolis Hastings, while the latter via the Metropolis-adjusted Langevin algorithm. To improve modularity and performance, each `Hierarchy` stores the 'unique' value τ_h and the observations $\boldsymbol{y}_h := \{y_i : c_i = h\}$ or, as it is often the case, the sufficient statistics of $\boldsymbol{y}_h$ needed to sample from the full conditional of τ_h. The implemented hierarchies at the time of writing are reported in Table 13.4.1.

The `Mixing` module Step 2 in Section 13.3.2 depends only on the prior on $\boldsymbol{w}$ and on the cluster allocations, while Step 1 in both Sections 13.3.1 and 13.3.2 requires an interaction between the weights (or the EPPF) and the hierarchies. Since the steps of the two algorithms are invariant to the choice of the prior for $\boldsymbol{w}$, we argue that this should be a further building block of the code. In our code, we represent a prior on $\boldsymbol{w}$ and the induced EPPF in a class called `Mixing`.

The following `Mixing` classes are currently available in the library:

1. `DirichletMixing`: it represents the EPPF of a Dirihclet Process (Ferguson, 1973),
2. `PitYorMixing`: it represents the EPPF of a Pitman-Yor Process (Pitman and Yor, 1997),
3. `TruncatedSBMixing`: the prior on $\boldsymbol{w}$ given by a truncated stick breaking process (Ishwaran and James, 2001),
4. `LogitSBMixing`: the *dependent* prior on $\boldsymbol{w}(x_i)$, x_i being a given covariate vector, as in Rigon and Durante (2021).
5. `MixtureFiniteMixing`: it represents the EPPF of a finite mixture with Dirichlet-distributed weights as in Miller and Harrison (2018).

The `Algorithm` module Finally, `Algorithm` classes are in charge of running the MCMC simulations. An `Algorithm` operates on a `Mixing` and several `Hierarchies` (or clusters), calling their appropriate update methods (and passing the appropriate data as input).

Of course, not every choice of `Mixing` and `Hierarchy` can be used in combination with all the choices of `Algorithm`. For instance, Neal's Algorithm 2 requires that the `Hierarchy` is conjugate, while the blocked Gibbs sampler requires m to be finite and fixed. Moreover, the EPPF might not be available analytically for all choices of `Mixing`. Nonetheless, we argue that these are consistent building blocks that allow us to exploit the structure shared by the algorithms without introducing redundant copy-pasted code.

Class Name	$f(\cdot \mid \tau)$	$G_0(\cdot)$	conjugate
`NNIGHierarchy`	$\mathcal{N}(\cdot \mid \mu, \sigma^2)$	$\mathcal{N}(\mu \mid \mu_0, \sigma^2/\lambda) IG(\sigma^2 \mid a, b)$	true
`NNxIGHierarchy`	$\mathcal{N}(\cdot \mid \mu, \sigma^2)$	$\mathcal{N}(\mu \mid \mu_0, \sigma_0^2) IG(\sigma^2 \mid a, b)$	false
`LapNIGHierarchy`	Laplace$(\cdot \mid \mu, \lambda)$	$\mathcal{N}(\mu \mid \mu_0, \sigma_0^2) IG(\lambda \mid a, b)$	false
`NNWHierarchy`	$\mathcal{N}_d(\cdot \mid \mu, \Sigma)$	$\mathcal{N}_d(\mu \mid \mu_0, \Sigma/\lambda) IW(\Sigma \mid \nu, \psi)$	true
`LinRegUniHierarchy`	$\mathcal{N}(\cdot \mid x^t\beta, \sigma^2)$	$\mathcal{N}_p(\beta \mid \beta_0, \sigma^2\Lambda^{-1}) IG(\sigma^2 \mid a, b)$	true
`FAHierarchy`	$\mathcal{N}_p(\cdot \mid \mu, \Sigma + \Lambda\Lambda^\top)$	$\mathcal{N}_p(\mu \mid \mu_0, \psi I)\mathrm{DL}(\Lambda \mid a)$ $\prod_{j=1}^p IG(\sigma_j^2 \mid a, b)$	false

Table 13.4.1: The hierarchies implemented in **BayesMix**. IG stands for the Inverse-Gamma distribution while DL for the Dirichlet-Laplace distribution (Bhattacharya et al., 2015).

Class Name	Reference	non-conjugate	marginal
`Neal2Algorithm`	Neal (2000)	false	true
`Neal3Algorithm`	Neal (2000)	false	true
`Neal8Algorithm`	Neal (2000)	true	true
`BlockedGibbsAlgorithm`	Ishwaran and James (2001)	true	false
`SplitAndMergeAlgorithm`	Jain and Neal (2004)	false	true

Table 13.4.2: The algorithms coded in **BayesMix**. From left to right: name of the class, bibliographic reference, indicator for accepting non-conjugate hierarchies, if the mixing must implement the *marginal* methods (true) or the *conditional* ones (false).

13.5 Hands on examples

Here we show how to install and use the **BayesMix** library. The section is meant for users who are not expert C++ programmers and only need to use what is already included in the library. See Section 13.7 for material aimed at more advanced users.

13.5.1 Installing the BayesMix library

We provide a handy `cmake` installation that automatically handles all the dependencies. After downloading the repository from Github, it is sufficient to build the executables using `cmake`. We provide detailed instructions below.

Unix-like machines On Unix-like machines (including those featuring macOS) it is sufficient to open the terminal and navigate to the `bayesmix` folder. Then the following commands

```
mkdir build
cd build
cmake ..
make run_mcmc
make plot_mcmc
```

create the executables `run_mcmc` and `plot_mcmc` inside the `build` directory.

Windows machines At this stage of development, Windows machines are supported only via Windows Subsystem for Linux (WSL). Hence, in order to build **BayesMix** on Windows, you simply need to follow the instructions for Unix-like machines from the Linux terminal.

13.5.2 Using the BayesMix library

There are two ways to interact with **BayesMix**. C++ users can create an executable linking against **BayesMix** or use (a possibly customized version of) the run_mcmc executable, which receives a list of command line arguments defining the model and the path to the data, runs the MCMC algorithm and writes the chains to a file. We give an example below. Alternatively, Python users can interact with **BayesMix** via the **bayesmixpy** interface. In both cases, we consider a Dirichlet process mixture of univariate normals, i.e.

$$
\begin{aligned}
y_1, \dots, y_n \mid \boldsymbol{w}, \boldsymbol{\tau} &\overset{\text{iid}}{\sim} \sum_{h=1}^{\infty} w_h \mathcal{N}(\mu_h, \sigma_h^2) \\
w_1 = \nu_1, \quad w_j &= \nu_j \prod_{\ell<j} (1 - \nu_j), \quad j > 1 \\
\nu_j &\overset{\text{iid}}{\sim} \text{Beta}(1, \alpha) \\
\tau_h := (\mu_h, \sigma_h^2) &\overset{\text{iid}}{\sim} \mathcal{N}(\mu_h \mid \mu_0, \sigma_h^2/\lambda)\, \mathcal{IG}(\sigma_h^2 \mid a, b)
\end{aligned}
\tag{13.5}
$$

13.5.2.1 An example via the command line

In our code, model (13.5) can be declared assuming that the mixing is the DirichletMixing class and the hierarchy is the NNIGHierarchy class. We will use algorithm Neal2 for posterior simulation. We declare the model using three text files. In dp_param.asciipb we fix the "total mass" parameter of the Dirichlet process (i.e., α in (13.5)) to be equal to 1.0.

```
fixed_value {
    totalmass: 1.0
}
```

In g0_param.asciipb we set the parameters of the Normal-Inverse-Gamma prior G_0 as $(\mu_0, \lambda, a, b) = (0.0, 0.1, 2.0, 2.0)$:

```
fixed_values {
    mean: 0.0
    var_scaling: 0.1
    shape: 2.0
    scale: 2.0
}
```

Finally, in algo_param.asciipb we specify the algorithm, the number of iterations (and burn-in), and the random seed as follows:

```
algo_id: "Neal2"
rng_seed: 20201124
iterations: 1500
burnin: 500
init_num_clusters: 3
```

To run the executable, we call the build/run_mcmc executable with the appropriate parameters:

```
build/run_mcmc \
  --algo-params-file algo_param.asciipb \
  --hier-type NNIG --hier-args g0_param.asciipb \
  --mix-type DP --mix-args dp_param.asciipb \
  --coll-name chains.recordio \
```

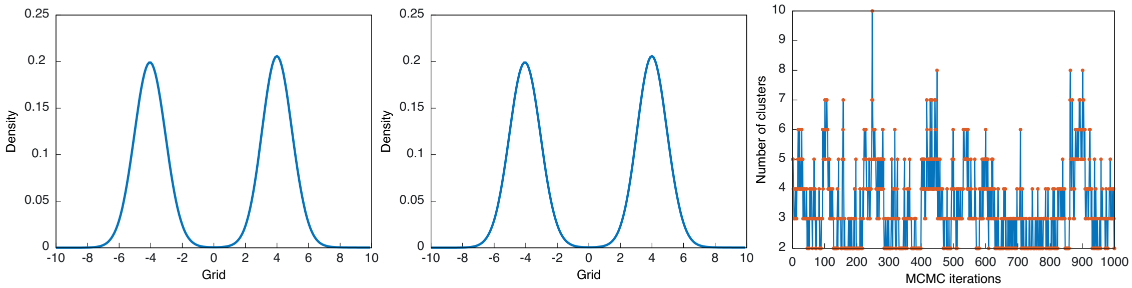


Figure 13.5.1: Plots from `plot_mcmc` executable: density estimate (left), histogram (center) and traceplot (right) of the number of clusters. The example refers to the `DirichletMixing` module described in Section 13.5.2.

```
   --data-file data.csv \
   --grid-file grid.csv \
   --dens-file eval_dens.csv \
   --n-cl-file numclust_chain.csv \
   --clus-file clustering_chain.csv \
   --best-clus-file best_clustering.csv
```

where the first command line arguments are used to specify the model and algorithm. In particular, the argument `---coll-name` specifies which collector to use. If it is not "`memory`", then the `FileCollector` (see Section 13.7.4) will be used and chains stored in the corresponding file. The remaining arguments consist of the path to the files containing the observations (`---data-file`), the grid where to evaluate the predictive density (`---grid-file`), and the files where to store the predictive (log) density (`---dens-file`), the MCMC chain of the number of clusters (`---n-cl-file`), the MCMC chain of the cluster allocation variables (`---clus-file`) and the best clustering obtained by minimizing the posterior expectation of Binder's loss function (`---best-clus-file`). If any of the arguments from `---grid-file` to `---best-clus-file` is empty, the computations required to get the associated quantities are skipped.

After the MCMC algorithm has finished to run and all the quantities of interest have been saved to `csv` files, it is easy to load them into another software program to summarize posterior inference through plots. For basic uses, we provide a self-contained executable named `plot_mcmc` which plots and saves the posterior predictive density (Figure 13.5.1, left panel), the posterior distribution of the number of clusters (Figure 13.5.1 (center panel)) and the traceplot of the number of clusters (Figure 13.5.1, right panel).

13.5.2.2 An example through the Python interface

As mentioned before, we also provide (**bayesmixpy**), a Python interface that does not require users to use the terminal. To install the **bayesmixpy** package, navigate to the `python` sub-folder and execute in the terminal "`python3 -m pip install -e .`". Once it is installed, the package provides the `build_bayesmix()` and `run_mcmc()` functions. The former installs the executable while the latter is used to run the MCMC chains. Below, we provide a hands-on example.

First, we build **BayesMix**:

```
from bayesmixpy import build_bayesmix, run_mcmc
build_bayesmix(nproc=4)
>>> ...
>>>  export the environment variable
        BAYESMIX_EXE=<BAYESMIX_PATH>/build/run_mcmc
```

Observe that the last output line specifies the location of the executable and asks users to export the environmental variable BAYESMIX_EXE. We can do it directly in Python as follows

```
import os
os.environ["BAYESMIX_EXE"] = "<BAYESMIX_PATH>/build/run_mcmc"
```

where <BAYESMIX_PATH>/build/run_mcmc is the path printed by build_bayesmix.

We are now ready to declare our model. We assume a DirichletMixing as mixing and a NNIGHierarchy as hierarchy. The following code snippet specifies that the "total mass" parameter of the Dirichlet process is fixed to 1.0, the parameters of the Normal-Inverse-Gamma prior are fixed to $(\mu_0, \lambda, a, b) = (0.0, 0.1, 2.0, 2.0)$ and we will run Neal2Algorithm for 1,500 iterations, discarding the first 500 as burn-in.

```
dp_params = """
fixed_value {
    totalmass: 1.0
}
"""

g0_params = """
fixed_values {
    mean: 0.0
    var_scaling: 0.1
    shape: 2.0
    scale: 2.0
}
"""

algo_params = """
    algo_id: "Neal2"
    rng_seed: 20201124
    iterations: 1500
    burnin: 500
    init_num_clusters: 3
"""
```

Finally, we run the MCMC algorithm on some simulated data, as simply as:

```
import numpy as np

data = np.concatenate([np.random.normal(size=100) - 3,
                       np.random.normal(size=100) + 3])
dens_grid = np.linspace(-6, 6, 1000)
log_dens, numcluschain, cluschain, bestclus = run_mcmc(
    "NNIG", "DP", data, go_params, dp_params, algo_params,
    dens_grid=dens_grid, return_clusters=True,
    return_num_clusters=True, return_best_clus=True)
```

which returns the log of the predictive density evaluated at dens_grid for each iteration of the MCMC sampling, the chain of the number of clusters, the chain of the cluster allocations, and the best clustering obtained by minimizing the posterior expectation of Binder's loss function. We summarize the inference in a plot as follows:

```
import matplotlib.pyplot as plt
```

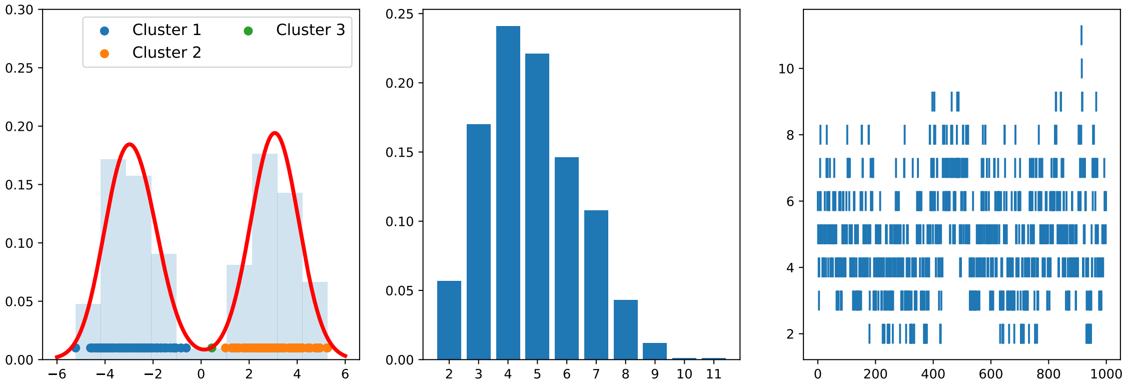


Figure 13.5.2: Output plot for the Python example: density estimate (left), histogram (center) and traceplot (right) of the number of clusters. The example refers to model (13.5) described in Section 13.5.2.

```
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(20, 5))

axes[0].hist(data, alpha=0.2, density=True)
for c in np.unique(bestclus):
    data_in_clus = data[bestclus == c]
    axes[0].scatter(
        data_in_clus, np.zeros_like(data_in_clus) + 0.01,
        label="Cluster {0}".format(int(c) + 1))
axes[0].plot(
    dens_grid, np.exp(np.mean(log_dens, axis=0)),
    color="red", lw=3)
axes[0].legend(fontsize=16, ncol=2, loc=1)
axes[0].set_ylim(0, 0.3)


x, y = np.unique(numcluschain, return_counts=True)
axes[1].bar(x, y / y.sum())
axes[1].set_xticks(x)

axes[2].vlines(np.arange(len(numcluschain)),
               numcluschain-0.3, numcluschain+0.3)
plt.show()
```

The output of the above code is displayed in Figure 13.5.2.

We also consider an example with bivariate datapoints, the `faithful` dataset, a well-known benchmark dataset for Bayesian density estimation and cluster detection. In this case, we assume that $f(\cdot \mid \tau)$ is the bivariate Gaussian density, with parameters $\tau = (\mu, \Psi = \Sigma^{-1})$ being the mean and precision matrix, respectively. A suitable prior for μ, Ψ is the Normal-Wishart distribution, i.e. $\mu \mid \Psi \sim \mathcal{N}_2(\mu_0, (\lambda\Psi)^{-1})$, $\Psi \sim IW(\nu_0, \Psi_0)$, with $\mathbb{E}(\Psi) = \Psi_0/(\nu - 2 - 1)$. To declare the model and run the MCMC algorithm, we can reuse most of the code of the univariate example, replacing the defintion of `g0_params` with:

```
g0_params = """
fixed_values {
    mean {
        size: 2
```

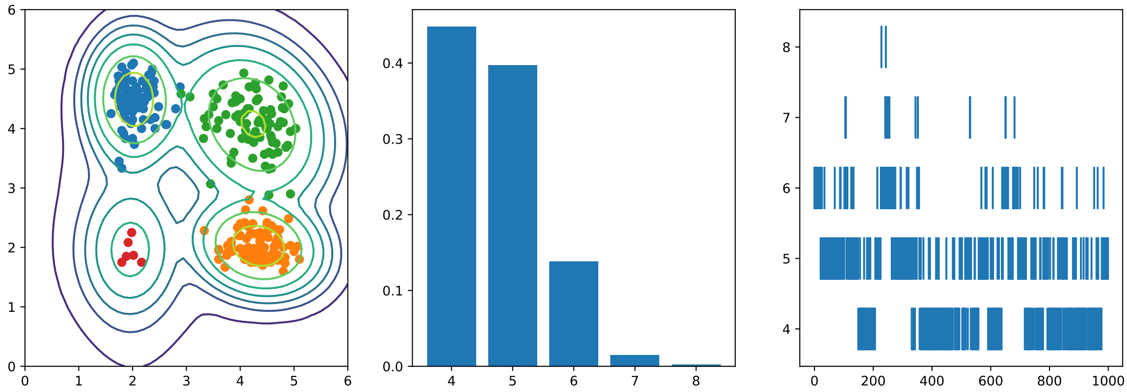


Figure 13.5.3: density estimate (left), histogram (center) and traceplot (right) of the number of clusters. The example refers to the `faithful` dataset in Section 13.5.2.

```
        data: [3.484, 3.487]
    }
    var_scaling: 0.01
    deg_free: 5
    scale {
        rows: 2
        cols: 2
        data: [1.0, 0.0, 0.0, 1.0]
        rowmajor: false
    }
}
"""
```

Posterior inference is summarized in Figure 13.5.3.

13.6 Performance benchmarking and comparisons

Here we compare the library **BayesMix** and the recently published **BNPmix** R package, which we have reviewed in Section 13.2, in terms of clustering quality and computational efficiency. All simulations were run on a Ubuntu 21.10 16 GB laptop machine. We consider three benchmark datasets for the comparison. The first two are the popular univariate `galaxy` and bivariate `faithful` datasets, both available in R. The third example is a simulated four-dimensional dataset, which we will refer to as `highdim`. It includes 10,000 points sampled from a Gaussian mixture with two equally weighted components, with mean $\mu_4 = [2, 2, 2, 2]$ and $-\mu_4$ respectively, and both covariance matrices equal to the identity matrix.

Since **BNPmix** focuses on Pitman-Yor processes and does not implement the Gamma prior for the total mass of the Dirichlet process, comparison is made using only Pitman-Yor mixtures with the same hyperparameter values for both libraries, including Pitman-Yor parameters and hierarchy hyperprior values. We test **BayesMix** using four different marginal algorithms – `Neal2`, `Neal3`, `Neal8`, and `SplitMerge`. The package **BNPmix** uses its own implementation of `Neal2`, which is referred to as `mar`, and the authors' newly implemented importance conditional sampler, or `ics` for short. Each algorithm has been run for 5,000 iterations, with 1,000 iterations as burn-in period.

Autocorrelation plots for the number of clusters for all runs are displayed in Figure 13.6.1. **BayesMix** algorithms show better mixing properties of the MCMC chain, particularly in the bivariate galaxy case, where **BNPmix** struggles to reduce to zero the autocorrelation for large lags.

As far as computational efficiency is concerned, we report Effective Sample Size (ESS), running times, and ESS-over-time ratio of the MCMC simulations for the above tests in Tables 13.6.1, 13.6.2, and 13.6.3. ESS measures the quality of a chain in terms of equivalent, hypothetical sample size of independent observations. All **BayesMix** algorithms perform much better than **BNPmix** ones in terms of ESS while achieving comparable or lower running times. Neal2, i.e. the same algorithm as **BNPmix**'s mar, and Neal3 stand out as being particularly efficient as quantified by the three metrics, especially as the datapoint dimension grows larger (faithful and highdim).

As a final example for this comparison, we have simulated ten-dimensional datapoints from a Gaussian mixture with two well- separated components (with equal weights). As for highdim, the sample size is 10,000. All algorithms in **BayesMix** but Neal2 have been able to correctly distinguish the two clusters, whereas **BNPmix** failed to do so, identifying only one. The four- and ten-dimensional examples show that **BayesMix** has a scalable approach that works even with large, high-dimensional datasets.

13.7 Topics for expert users

The goal of this section is to give an example on how new users can extend the library by implementing a new Mixing or Hierarchy. To do so, the C++ code structure and the APIs of each base class must be explained in greater detail.

We give more details on the main building blocks in **BayesMix**. We follow an object-oriented approach and we adopt a combination of runtime and compile-time polymorphism based on inheritance and templates, using the so called curiously recurring template pattern (CRTP), as explained in Sections 13.7.1 and 13.7.2.

13.7.1 The Mixing module

As previously mentioned, a Mixing represents the prior distribution over the weights $\boldsymbol{w}$ and the associated EPPF. The AbstractMixing class defines the following API:

```
class AbstractMixing {
 public:
  virtual void initialize() = 0;

  virtual double get_mass_existing_cluster(
      const unsigned int n, const bool log, const bool propto,
      std::shared_ptr<AbstractHierarchy> hier,
      const Eigen::RowVectorXd &covariate=
        Eigen::RowVectorXd(0));

  virtual double get_mass_new_cluster(
      const unsigned int n, const bool log, const bool propto,
      const unsigned int n_clust,
      const Eigen::RowVectorXd &covariate=
        Eigen::RowVectorXd(0));

  virtual Eigen::VectorXd get_mixing_weights(
      const bool log, const bool propto,
      const Eigen::RowVectorXd &covariate=
        Eigen::RowVectorXd(0));
```

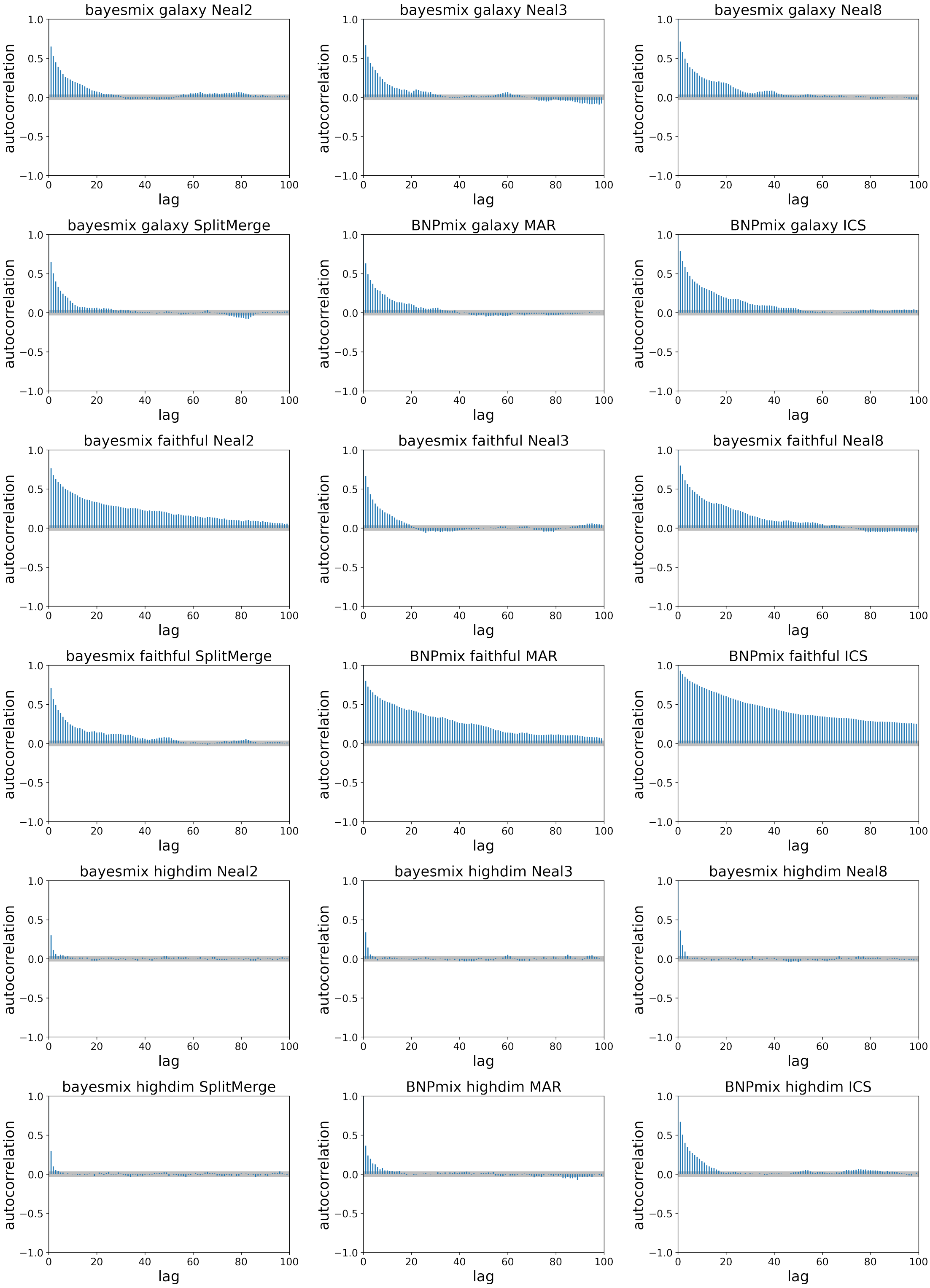


Figure 13.6.1: Comparison between autocorrelation plots on the number of clusters of the `galaxy` (top two rows), `faithful` (middle two rows), and `highdim` (bottom two rows) datasets

	algorithm	ESS	time	ESS/time
BNPmix	`mar`	338.562	0.827	409.469
	`ics`	162.128	0.842	192.438
BayesMix	`Neal2`	337.467	0.370	912.073
	`Neal3`	340.332	0.611	557.009
	`Neal8`	191.580	0.589	325.263
	`SplitMerge`	400.551	1.218	328.860

Table 13.6.1: Comparison of metrics for the `galaxy` dataset

	algorithm	ESS	time	ESS/time
BNPmix	`mar`	36.288	3.733	9.721
	`ics`	15.499	1.949	7.954
BayesMix	`Neal2`	80.648	1.823	44.239
	`Neal3`	394.709	4.796	82.300
	`Neal8`	139.419	5.746	24.264
	`SplitMerge`	217.788	12.278	17.738

Table 13.6.2: Comparison of metrics for the `faithful` dataset

	algorithm	ESS	time	ESS/time
BNPmix	`mar`	978.471	1063.740	0.920
	`ics`	426.749	47.084	9.064
BayesMix	`Neal2`	1578.956	44.866	35.193
	`Neal3`	1861.819	166.151	11.206
	`Neal8`	1617.569	296.635	5.453
	`SplitMerge`	1865.773	870.494	2.143

Table 13.6.3: Comparison of metrics for the `highdim` dataset

```
  virtual void update_state(
      const std::vector<std::shared_ptr<AbstractHierarchy>>
            &unique_values,
      const std::vector<unsigned int> &allocations) = 0;
};
```

In addition to these methods, AbstractMixing defines input-output functionalities discussed in Section 13.7.4.

The get_mass_existing_cluster() and get_mass_new_cluster() methods evaluate the EPPF Φ. Specifically, get_mass_existing_cluster() evaluates $\Phi(n_1, \ldots, n_h + 1, \ldots, n_k) = f_1(n_h + 1, n, \theta)$ for a given h, while get_mass_new_cluster() evaluates $\Phi(n_1, \ldots, n_h, \ldots, n_k{+}1) = f_2(k, n, \theta)$ as defined in (13.4). Instead, get_mixing_weights() returns the vector of weights $\boldsymbol{w}$. Both methods used to evaluate the EPPF take as input the number n of observations in the model, as well as two boolean flags (propto, log) specifying if the result must be returned up to a proportionality constant and in log-scale. The get_mass_existing_cluster() method also receives a pointer to the Hierarchy the cluster represents. Note that the three methods take as input a vector of covariates, which is the empty vector by default and can be used to define *dependent* mixture models, for instance, by assuming the dependency logit stick breaking prior implemented in LogitSBMixing.

The update_state() method allows the child classes to assume hyperpriors on all the parameters. The update_state() method is used to sample parameters $\boldsymbol{w}, m$ and additional hyperparameters from their full conditional.

Child classes do not inherit directly from AbstractMixing, but rather from a template class which in turn inherits from AbstractMixing, in the following way:

```
template <class Derived, typename State, typename Prior>
class BaseMixing : public AbstractMixing {
...
}
```

The BaseMixing class allows for more flexible code since it is templated over two objects representing the State and the Prior. For instance, in the case of a Pitman-Yor process, the state is defined as:

```
namespace PitYor {
  struct State {
    double strength, discount;
  };
};
```

but more complex objects can be used as well. Moreover, BaseMixing implements several virtual methods from the AbstractMixing class, so that end users only need to focus on the code that is specific to a given model. For instance, a *marginal* mixing such as DirichletProcess only needs to implement the following methods:

```
void update_state(
      const std::vector<std::shared_ptr<AbstractHierarchy>>
            &unique_values,
      const std::vector<unsigned int> &allocations) override;

double mass_existing_cluster(
    const unsigned int n, const bool log, const bool propto,
    std::shared_ptr<AbstractHierarchy> hier) const override;
```

```
double mass_new_cluster(
    const unsigned int n, const bool log, const bool propto,
    const unsigned int n_clust) const override;
```

and some input-output functionalities. Instead, a *conditional* mixing such as TruncatedSBMixing implements the following functions:

```
void update_state(
      const std::vector<std::shared_ptr<AbstractHierarchy>>
            &unique_values,
      const std::vector<unsigned int> &allocations) override;

Eigen::VectorXd get_weights(
    const bool log, const bool propto) const override;
```

13.7.2 The Hierarchy module

The Hierarchy module represents the Bayesian model

$$\begin{aligned} y_j \mid \tau &\overset{\text{iid}}{\sim} f(\cdot \mid \tau), \quad j = 1, \dots, l \\ \tau &\sim G_0 \end{aligned} \tag{13.6}$$

Where $f(\cdot \mid \cdot)$ is the mixture component and G_0 the base measure. Given the model (13.6), we are interested in: (*i*) evaluating the (log) likelihood function $f(x \mid \tau)$ for a given x, (*ii*) sampling from the prior model $\tau \sim G_0$, and (*iii*) sampling from the full conditional of $\tau \mid y_1, \dots, y_\ell$. Each of these goals is delegated to a different class, namely the Likelihood, the PriorModel, and the Updater. Then a Hierarchy class is in charge of making Likelihood, PriorModel, and Updater communicate with each other and provides a common API for all possible models.

The choice of separating Likelihood, PriorModel, and Updater allows for great flexibility. In fact, we could have different Hierarchy classes that employ the same Likelihood but a different PriorModel. Moreover, different Updaters can be used. If the model is conjugate or semi-conjugate, a specific SemiConjugateUpdater is usually preferred. If this is not the case, we provide off-the-shelf RandomWalkUpdater and MALAUpdater that implement a random-walk Metropolis-Hastings move or a Metropolis-adjusted Langevin algorithm move, which can be used for any combination of Likelihood and PriorModel. As a consequence, users do not need to code an Updater if they want to implement a new model.

Throughout this section, we consider the illustrative example where $\tau = (\mu, \sigma^2)$, $f(\cdot \mid \tau) = \mathcal{N}(\cdot \mid \mu, \sigma^2)$ is the univariate Gaussian density and $G_0(\mu, \sigma^2) = \mathcal{N}(\mu \mid \mu_0, \sigma^2/\lambda) IG(\sigma^2 \mid a, b)$ is the Normal-inverse-Gamma distribution.

The Hierarchy module and all its sub-modules (Likelihood, PriorModel, State and Updater) achieve runtime polymorphism through an abstract interface (which establishes which operations can be performed by the end user) and employing the Curiously Recurring Template Pattern (CRTP Coplien, 1995).

Let us explain the structure in more detail, starting with the Hierarchy module. First, an AbstractHierarchy defines the following API:

```
class AbstractHierarchy {
 public:
  double get_like_lpdf(
      const Eigen::RowVectorXd &datum,
      const Eigen::RowVectorXd &covariate) const;
  virtual void sample_prior() = 0;

  virtual void sample_full_cond(bool update_params) = 0;

  virtual void add_datum(
      const int id, const Eigen::VectorXd &datum,
      const bool update_params,
      const Eigen::VectorXd &covariate) = 0;

  virtual void remove_datum(
      const int id, const Eigen::VectorXd &datum,
      const bool update_params,
      const Eigen::VectorXd &covariate)) = 0;
};
```

In the code above, `get_like_lpdf()` evaluates the likelihood function $f(y \mid \tau)$ for a given datapoint, `sample_prior()` samples from G_0, and `add_datum()` (`remove_datum()`) are called when allocating (removing) a datum from the current cluster.

As in the case of `Mixings`, child classes inherit from a template class with respect to the `Likelihood` and the `PriorModel` from the `BaseHierarchy` class. Most of the methods in the API are implemented in this class. Thus, coding a new hierarchy is extremely simple within this framework, since only very few methods need to be implemented from scratch. All the hierarchies available so far inherit from this class and are reported in Table 13.4.1.

13.7.2.1 The `Likelihood` sub-module

The `Likelihood` sub-module represents the likelihood we have assumed for the data in a given cluster. Each `Likelihood` class represents the sampling model

$$y_1, \ldots, y_k \mid \boldsymbol{\tau} \overset{\text{iid}}{\sim} f(\cdot \mid \boldsymbol{\tau})$$

for a specific choice of the probability density function f.

In principle, the `Likelihood` classes are responsible only of evaluating the log-likelihood function given a specific choice of parameters $\boldsymbol{\tau}$. Therefore, a simple inheritance structure would seem appropriate. However, the nature of the parameters $\boldsymbol{\tau}$ can be very different across different models (think for instance of the difference between the univariate normal and the multivariate normal paramters). As such, we again employ CRTP to manage the polymorphic nature of `Likelihood` classes.

The `AbstractLikelihood` class provides the following common API:

```
class AbstractLikelihood {
 public:
  double lpdf(
      const Eigen::RowVectorXd &datum,
      const Eigen::RowVectorXd &covariate =
            Eigen::RowVectorXd(0)) const;

  virtual Eigen::VectorXd lpdf_grid(
      const Eigen::MatrixXd &data,
      const Eigen::MatrixXd &covariates =
            Eigen::MatrixXd(0, 0)) const = 0;

  virtual double cluster_lpdf_from_unconstrained(
      Eigen::VectorXd unconstrained_params) const;

  virtual stan::math::var cluster_lpdf_from_unconstrained(
      Eigen::Matrix<stan::math::var, Eigen::Dynamic, 1>
            unconstrained_params) const;

  virtual bool is_multivariate() const = 0;

  virtual bool is_dependent() const = 0;

  virtual void add_datum(
      const int id, const Eigen::RowVectorXd &datum,
      const Eigen::RowVectorXd &covariate =
            Eigen::RowVectorXd(0)) = 0;

  virtual void remove_datum(
      const int id, const Eigen::RowVectorXd &datum,
      const Eigen::RowVectorXd &covariate =
            Eigen::RowVectorXd(0)) = 0;


  void update_summary_statistics(
      const Eigen::RowVectorXd &datum,
      const Eigen::RowVectorXd &covariate, bool add);

  virtual void clear_summary_statistics() = 0;
};
```

First of all, we require the implementation of the `lpdf()` and `lpdf_grid()` methods, which simply evaluate the loglikelihood in a given point or in a grid of points (also in case of a *dependent* likelihood, i.e., in which covariates are associated to each observation). The `cluster_lpdf_from_unconstrained()` method allows the evaluation of the likelihood of the whole cluster starting from the vector of unconstrained parameters. This is a key method which is only needed if a Metropolis-like updater is used. Observe that the `AbstractLikelihood` class provides two such methods, one returning a `double` and one returning a `stan::math::var`. The latter is used to automatically compute the gradient of the likelihood via Stan's automatic differentiation, if needed. In practice, users do not need to implement both methods separately, and can implement only one templated method; see the `UniNormLikelihood` example below. The `add_datum()` and `remove_datum()` methods manage the insertion and deletion of a data point in the given cluster, and update the summary statistics associated with the likelihood using the `update_summary_statistics()` method. Summary statistics (when available) are used to evaluate the likelihood function on the whole cluster, as well as to perform the posterior updates of $\boldsymbol{\tau}$. This usually gives a substantial speed-up.

Given this API, we define the `BaseLikelihood` class, which is a template class with respect to itself (thus enabling CRTP) and a `State`. The latter is a class which stores the parameters $\boldsymbol{\tau}$ and eventually manages the transformation in its unconstrained form (for Metropolis updaters), if any. The `BaseLikelihood` class is declared as follows:

```
template <class Derived, typename State>
    class BaseLikelihood : public AbstractLikelihood
```

This class implements methods that are common to all the likelihoods, in order to minimize the code that end users need to implement. Note that every concrete implementation of a likelihood model inherits from such a class. The following likelihoods are currently implemented in **BayesMix**:

1. `UniNormLikelihood`, that is $y \mid \mu, \sigma^2 \sim \mathcal{N}(\mu, \sigma^2)$, $\mu \in \mathbb{R}$, $\sigma^2 > 0$.
2. `MultiNormLikelihood`, that is $y \mid \mu, \Sigma \sim \mathcal{N}_d(\mu, \Sigma)$, $\mu \in \mathbb{R}^d$, Σ a symmetric and positive definite covariance matrix.
3. `FALikelihood`, that is $y \mid \mu, \Sigma \sim \mathcal{N}_d(\mu, \Sigma + \Lambda\Lambda^\top)$, $\mu \in \mathbb{R}^d$, $\Sigma = \mathrm{diag}(\sigma_1^2, \dots, \sigma_d^2)$, $\sigma_j^2 > 0$, Λ a $d \times p$ matrix (usually $p \ll d$, hence the name factor-analyzer likelihood).
4. `LinRegUniLikelihood`, that is $y \mid \beta, \sigma^2 \sim \mathcal{N}(x^\top \beta, \sigma^2)$, $\beta \in \mathbb{R}^d$, $\sigma > 0$. Here x is a vector of covariates, meaning that this hierarchy is *dependent*.
5. `UniLapLikelihood`, that is $y \mid \mu, \lambda \sim \mathrm{Laplace}(\mu, \lambda)$, $\mu \in \mathbb{R}$, $\lambda > 0$.

We report the code for `UniNormLikelihood` as an illustrative example:

```
class UniNormLikelihood
     : public BaseLikelihood<UniNormLikelihood, State::UniLS> {
  public:
   UniNormLikelihood() = default;

   ~UniNormLikelihood() = default;

   bool is_multivariate() const override { return false; };

   bool is_dependent() const override { return false; };

   void clear_summary_statistics() override;

   template <typename T>
   T cluster_lpdf_from_unconstrained(
       const Eigen::Matrix<T, Eigen::Dynamic, 1>
            &unconstrained_params) const;

  protected:
   double compute_lpdf(
        const Eigen::RowVectorXd &datum) const override;

   void update_sum_stats(
        const Eigen::RowVectorXd &datum, bool add) override;

   double data_sum = 0;

   double data_sum_squares = 0;
 };
```

13.7.2.2 The `PriorModel` sub-module

This sub-module represents the prior for the parameters in the likelihood, i.e.

$$\boldsymbol{\tau} \sim G_0$$

with G_0 being a suitable prior on the parameters space. We also allow for more flexible priors adding further level of randomness (i.e. the hyperprior) on the parameter characterizing G_0. Similarly to the case of `Likelihood` sub-module, we need to rely on a design pattern that can manage a wide variety of specifications. We rely once more on the CRTP approach, thus defining an API via a pure virtual class: `AbstractPriorModel`, which collects the methods each class should implement. This class is defined as follows:

```
class AbstractPriorModel {
 public:

  virtual double lpdf(
    const google::protobuf::Message &state_) = 0;

  virtual double lpdf_from_unconstrained(
      Eigen::VectorXd unconstrained_params) const;

  virtual stan::math::var lpdf_from_unconstrained(
      Eigen::Matrix<stan::math::var,Eigen::Dynamic,1>
        unconstrained_params) const;

  virtual std::shared_ptr<google::protobuf::Message> sample(
      ProtoHypersPtr hier_hypers = nullptr) = 0;

  virtual void update_hypers(
      const std::vector<bayesmix::AlgorithmState::ClusterState>
            &states) = 0;
};
```

The `lpdf()` and `lpdf_from_unconstrained()` methods evaluate the log-prior density function at the current state $\boldsymbol{\tau}$ or its unconstrained representation. In particular, `lpdf_from_unconstrain` is needed by Metropolis-like updaters; see below for further details. The `sample()` method generates a draw from the prior distribution. If `hier_hypers` is `nullptr`, the prior hyperparameter values are used. To allow sampling from the full conditional distribution in case of semi-congugate hierarchies, we introduce the `hier_hypers` parameter, which is a pointer to a `Protobuf` message storing the hierarchy hyperaprmeters to use for the sampling. The `update_hypers()` method updates the prior hyperparameters, given the vector of all cluster states.

Given the API, we define the `BasePriorModel` class, which is declared as:

```
template <class Derived, class State,
          typename HyperParams, typename Prior>
class BasePriorModel : public AbstractPriorModel
```

Such a class is derived from `AbstractPriorModel`. It is a template class with respect to itself (for CRTP), a `State` class (which represents the parameters over which the prior is assumed) an `HyperParams` type (which is a simple struct that codes the parameters characterizing G_0) and a `Prior` (which codes hierarchical priors for the G_0 parameters for more flexible and robust prior models). Like in previous sub-modules, this class manages code exceptions and implements general methods. Every concrete implementation of a prior model must be defined as an inherited class of `BasePriorModel`. The library currently supports the following priors:

1. `NIGPriorModel` $\mu \,|\, \sigma^2 \sim \mathcal{N}(\mu_0, \sigma^2/\lambda)$, $\sigma^2 \sim IG(a, b)$.

2. `NxIGPriorModel` $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$, $\sigma^2 \sim IG(a, b)$.

3. NWPriorModel $\mu \,|\, \Sigma \sim \mathcal{N}(\mu_0, \Sigma/\lambda)$, $\Sigma \sim IW(\nu_0, \Psi_0)$.
4. MNIGPriorModel $\beta \,|\, \sigma^2 \sim N_p(\mu, \sigma^2\Lambda^{-1})$, $\sigma^2 \sim IG(a, b)$
5. FAPriorModel $\mu \sim \mathcal{N}_p(\widetilde{\mu}, \psi I)$, $\Lambda \sim \mathrm{DL}(\alpha)$, $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$, $\sigma_j \overset{\text{iid}}{\sim} IG(a, b)$, $j = 1, \ldots, p$, where DL is the Dirichlet-Laplace distribution in Bhattacharya et al. (2015).

As an example, we report the implementation of the NIGPriorModel here below:

```
class NIGPriorModel : public BasePriorModel<
    NIGPriorModel, State::UniLS, Hyperparams::NIG,
    bayesmix::NNIGPrior> {
 public:
  using AbstractPriorModel::ProtoHypers;

  using AbstractPriorModel::ProtoHypersPtr;

  NIGPriorModel() = default;

  ~NIGPriorModel() = default;

  double lpdf(const google::protobuf::Message &state_) override;

  template <typename T>
  T lpdf_from_unconstrained(
      const Eigen::Matrix<T, Eigen::Dynamic, 1>
        &unconstrained_params) const;

  State::UniLS sample(ProtoHypersPtr hier_hypers=nullptr);

  void update_hypers(
    const std::vector<bayesmix::AlgorithmState::ClusterState>
        &states) override;

  void set_hypers_from_proto(
      const google::protobuf::Message &hypers_) override;

  std::shared_ptr<bayesmix::AlgorithmState::HierarchyHypers>
        get_hypers_proto() const override;

 protected:
  void initialize_hypers() override;
};
```

13.7.2.3 The Updater sub-module

The Updater module implements the machinery to provide a sampling from the full conditional distribution of a given hierarchy. Again, we rely on CRTP and define the API in the AbstractUpdater class as follows:

```
class AbstractUpdater {
 public:
  virtual bool is_conjugate() const;

  virtual void draw(
    AbstractLikelihood &like, AbstractPriorModel &prior,
    bool update_params) = 0;s
};
```

Here `is_conjugate()` declares whether the updater is meant to be used for a semi-conjugate hierarchy. The `draw` method is the key method of every updater: it receives `like` and `prior` as input, and updates the `State` (which is stored inside the `Likelihood`) by sampling it from conditional distribution $\boldsymbol{\tau} \mid y_1, \ldots, y_h$, where the y_j's are the data associated to one specific cluster. As already mentioned, when (13.6) is semi-conjugate, problem-specific updaters can be easily implemented by inheriting from the `SemiConjugateUpdater`; see, for instance, the code below.

```
class NNIGUpdater: public
        SemiConjugateUpdater<UniNormLikelihood, NIGPriorModel> {
  public:
   NNIGUpdater() = default;
   ~NNIGUpdater() = default;

   bool is_conjugate() const override { return true; };

   ProtoHypers compute_posterior_hypers(
        AbstractLikelihood &like,
        AbstractPriorModel &prior) override;
};
```

In particular, note that this class does not implement any `draw()` method. In fact, since the model is semi-conjugate, we exploit the `PriorModel` draw function but using updated parameters, which are computed by the `compute_posterior_hypers()` method.

If the model is not semi-conjugate, we suggest using `RandomWalkUpdater` or `MALAUpdater`, which sample from the full conditional distribution of $\boldsymbol{\tau}$ using a Metropolis-Hastings move. In this case, the following methods must be implemented in the `Likelihood` class:

```
template <typename T>
T cluster_lpdf_from_unconstrained(
    const Eigen::Matrix<T, Eigen::Dynamic, 1>
        &unconstrained_params) const;
```

while the prior should implement the following:

```
template <typename T>
T lpdf_from_unconstrained(
    const Eigen::Matrix<T, Eigen::Dynamic, 1>
        &unconstrained_params) const;
```

For instance, when f is the univariate Gaussian density, the unconstrained parameters are $(\mu, \log(\sigma^2))$. To evaluate the likelihood, it is sufficient to transform $\log(\sigma^2)$ using the exponential function. Instead, to evaluate the prior, one should take care of the correction in the density function due to the change of variables.

13.7.2.4 The `State` sub-module

`States` are classes used to store parameters τ_h's of every mixture component. Their main purpose is to handle serialization and de-serialization of the state; see also Section 13.7.4.

Moreover, they allow to go from the constrained to the unconstrained representation of the parameters (and viceversa) and compute the associated determinant of the Jacobian appearing in the change of density formula. All states inherit from a BaseState:

```
class BaseState {
 public:
  int card;
  using ProtoState = bayesmix::AlgorithmState::ClusterState;

  virtual Eigen::VectorXd get_unconstrained() {
    throw std::runtime_error("...");
  }
  virtual void set_from_unconstrained(const Eigen::VectorXd &in) {
    throw std::runtime_error("..."); }
  virtual double log_det_jac() { throw std::runtime_error("..."); }

  virtual void set_from_proto(
    const ProtoState &state_, bool update_card) = 0;
  virtual ProtoState get_as_proto() const = 0;
  std::shared_ptr<ProtoState> to_proto() const {
    return std::make_shared<ProtoState>(get_as_proto());
  }
};
```

Depending on the chosen Updater, the methods get_unconstrained(), set_from_unconstrained() and log_det_jac() might never be called. Therefore, we do not force users to implement them. Instead, the set_from_proto() and get_as_proto() are fundamental as they allow the interaction with Google's Protocol Buffers library; see Section 13.7.4 for more detail.

13.7.3 The Algorithm module

Mixing and Hierarchy classes are combined together by an Algorithm. Algorithms are direct implementation of MCMC samplers, such as Neal's Algorithm 2/3/8 and the blocked Gibbs sampler from Ishwaran and James (2001). All algorithms must inherit from the BaseAlgorithm class:

```
class BaseAlgorithm {
 protected:
  Eigen::MatrixXd data;
  Eigen::MatrixXd hier_covariates;
  Eigen::MatrixXd mix_covariates;

  std::vector<unsigned int> allocations;
  std::vector<std::shared_ptr<AbstractHierarchy>> unique_values;
  std::shared_ptr<BaseMixing> mixing;

  virtual void sample_allocations() = 0;
  virtual void sample_unique_values() = 0;

  virtual void step() {}

 public:
  void run(BaseCollector *collector);
  virtual Eigen::MatrixXd eval_lpdf(
      BaseCollector *const collector,
      const Eigen::MatrixXd &grid,
      const Eigen::MatrixXd &hier_covariates,
      const Eigen::MatrixXd &mix_covariates) = 0;
};
```

The `Algorithm` class saves the data and (optionally) two set of covariates: `hier_covariates` and `mix_covariates`. Therefore, it is trivial to extend the code to more general models to accommodate for covariate-dependent likelihoods and/or mixings. Moreover, the `Algorithm` also stores the cluster allocation variables (`allocations`), the hierachies representing the mixture components (`unique_values`) and the mixing (`mixing`). The last two objects are stored through pointers to the corresponding base class, to achieve runtime polymorphism.

The basic method from `Algorithm` is `step()` which performs a Gibbs sampling step calling the appropriate update methods for all the blocks of the model. A `run()` method is used to run the MCMC chain, i.e. `run()` calls `step()` for a user-specified number of iterations, possibly discarding an initial burn-in phase. The goal of MCMC simulations is to *collect* samples from the posterior distribution, which must be stored for later use. Hence, the `run()` receives as input an instance of `BaseCollector` which is indeed in charge of storing the visited states either in memory (RAM) or by saving in a file; see Section 13.7.4 for further details.

Since one of the main goals of mixture analysis is density estimation, an `Algorithm` must be also able to evaluate the mixture density on a fixed grid, given the visited samples. This is achieved by the `eval_lpdf()` method.

All the algorithms implemented in **BayesMix** are listed in Table 13.4.2.

13.7.4 I/O and cross-language functionalities

There is a final building block of **BayesMix**, that is the management of input / output (I/O). Most of C++ based packages for Bayesian inference, such as **Stan** (Stan Development Team, 2019) and **JAGS** (Plummer, 2017), rely on tabular formats to save the chains. Specifically, the output of an MCMC algorithm is collected in an array where each parameter is saved in a different column and the resulting object is then serialized in a text format (such as csv). This approach is simple but rather restrictive, since it requires a fixed number of parameters, which is usually not our case. Moreover, in case of non-scalar parameters (such as covariance matrices), these parameters need to be first *flattened* to be stored in a matrix and then they need to be re-built from this flattened version to compute posterior inference.

Instead, we rely on the powerful serialization library Protocol Buffers (https://developers.google.com/protocol-buffers/) to handle I/O operations. Specifically, this requires defining so-called *messages* in a `.proto` file. Semantically, the declaration of a message is alike the declaration of a C++ struct. For instance the following code:

```
message UniLSState {
  double mean = 1;
  double var = 2;
}
```

defines a message named `UniLSState` whose fields are two `double`s, `mean` and `var`. In more complex settings, other `Protobuf` messages can act as types for these variables. The `protoc` compiler operates on these messages and transpiles them into files implementing associated classes (one per message) in a given programming language (for us, it is of course C++). Then, the runtime library `google/protobuf` can be used to serialize and deserialize these messages very efficiently. All messages are declared in files placed in the `proto` folder. The transpilation into the corresponding C++ classes occurs automatically when installing the **BayesMix** library.

The state of the Markov chain can be stored in the following message:

```
message AlgorithmState {
  repeated ClusterState cluster_states = 1;
  repeated int32 cluster_allocs = 2 [packed = true];
  MixingState mixing_state = 3;
  int32 iteration_num = 4;
  HierarchyHypers hierarchy_hypers = 5;
}
```

where `ClusterState`, `MixingState` and `HierarchyHypers` are other messages defined in the `proto` folder.

In our code, there are classes that are exclusively dedicated to storing the samples from the MCMC, either in memory or on file. These are called `Collectors` and inherit from `BaseCollector` that defines the API:

```
class BaseCollector {
 public:
  virtual void start_collecting() = 0;

  virtual void finish_collecting() = 0;

  bool get_next_state(google::protobuf::Message *out);

  virtual void collect(
    const google::protobuf::Message &state) = 0;

  virtual void reset() = 0;

  unsigned int get_size() const;
```

A collector stores the entire MCMC chain in a data structure that resembles a linked list, that is, the collector knows the beginning of the chain and the current state. The function `get_next_state()` can be used to advance to the next state, while writing its values to a pointer. Instead, the algorithm calls the `collect()` method when a MCMC iteration must be saved.

13.7.5 Extending the BayesMix library

In this section, we show a concrete example of an extension of **BayesMix**. We consider a mixture model with $\mathrm{Gamma}(\cdot \mid \alpha, \beta)$ kernel, where α is a fixed parameter, and the mixing measure over β is a Dirichlet process with conjugate $\mathrm{Gamma}(\alpha_0, \beta_0)$ base measure. We can use any of the algorithms in **BayesMix** to sample from the posterior of this model, but we need to implement additional code in our library.

Three or four classes are needed: (i) a `GammaLikelihood` class representing a Gamma likelihood, (ii) a `GammaPriorModel` class representing a Gamma prior over the τ_h's, and (iii) a `GammaHierarchy` that combines `GammaLikelihood` and `GammaPriorModel`. As far as the updater is concerned, we could either use a `MetropolisUpdater` or implement a (iv) `GammaGammaUpdater` class that takes advantage of the conjugacy. In this example, we opt for the latter.

We will not cover in full detail the implementation of all the required functions, but just the core ones. The full code for this example is available at https://github.com/bayesmix-dev/bayesmix/tree/master/examples.

Since the state of each component is just (α, β_h), where α is fixed in our case, we can use the Protobuf message bayesmix::AlgorithmState::ClusterState::general_state to save it. That is, we save each (α, β_h) in a Vector of length two. This is done in the geta_as_proto() function implemented below. For more complex hierarchies, we suggest users to create their own Protobuf messages and add them to the bayesmix::AlgorithmState::ClusterS field.

We report the code for the State and GammaLikleihood classes below:

```
namespace State { class Gamma: public BaseState {
 public:
  double shape, rate;
  using ProtoState = bayesmix::AlgorithmState::ClusterState;

  ProtoState get_as_proto() const override {
    ProtoState out;
    out.mutable_general_state()->set_size(2);
    out.mutable_general_state()->mutable_data()->Add(shape);
    out.mutable_general_state()->mutable_data()->Add(rate);
    return out;
  }

  void set_from_proto(
        const ProtoState &state_, bool update_card) override {
    if (update_card) { card = state_.cardinality(); }
    shape = state_.general_state().data()[0];
    rate = state_.general_state().data()[1];
  }
};}

 class GammaLikelihood:
    public BaseLikelihood<GammaLikelihood, State::Gamma> {
 public:
  ...
  void clear_summary_statistics() override;

 protected:
  double compute_lpdf(
    const Eigen::RowVectorXd &datum) const override;
  void update_sum_stats(
    const Eigen::RowVectorXd &datum, bool add) override;

  double data_sum = 0;
  int ndata = 0;
};

void GammaLikelihood::clear_summary_statistics() {
  data_sum = 0;
  ndata = 0;
}
```

```
double GammaLikelihood::compute_lpdf(
    const Eigen::RowVectorXd &datum) const {
  return stan::math::gamma_lpdf(datum(0), state.shape, state.rate);
}

void GammaLikelihood::update_sum_stats(
    const Eigen::RowVectorXd &datum, bool add) {
  if (add) {
    data_sum += datum(0);
    ndata += 1;
  } else {
    data_sum -= datum(0);
    ndata -= 1;
  }
}
```

Next, we report the code for the GammaPriorModel class. As we did for the GammaLikelihood, we do not need to write any additional Protobuf messages. Instead, we rely on the HierarchyHypers::general_state field which saves the hyperparameters α_0 and β_0 in a Vector.

```
namespace Hyperparams {
 struct Gamma {
   double rate_alpha, rate_beta;
 };
}

class GammaPriorModel: public BasePriorModel<
        GammaPriorModel, State::Gamma, Hyperparams::Gamma,
        bayesmix::EmptyPrior> {
 public:
  using AbstractPriorModel::ProtoHypers;
  using AbstractPriorModel::ProtoHypersPtr;

  GammaPriorModel(double shape_=-1, double rate_alpha_=-1,
                  double rate_beta_=-1);
  ~GammaPriorModel() = default;

  double lpdf(
    const google::protobuf::Message &state_) override;

  State::Gamma sample(
    ProtoHypersPtr hier_hypers=nullptr) override;

  void update_hypers(
    const std::vector<bayesmix::AlgorithmState::ClusterState>
        &states) override {
    return;
  };

  void set_hypers_from_proto(
     const google::protobuf::Message &hypers_) override;
  ProtoHypersPtr get_hypers_proto() const override;
  double get_shape() const { return shape; };

 protected:
  double shape, rate_alpha, rate_beta;
  void initialize_hypers() override;
};

/* DEFINITIONS */
....
```

Finally, we implement a dedicated Updater as follows.

```
class GammaGammaUpdater: public
    SemiConjugateUpdater<GammaLikelihood, GammaPriorModel> {
  public:
   GammaGammaUpdater() = default;
   ~GammaGammaUpdater() = default;

   bool is_conjugate() const override { return true; };

   ProtoHypersPtr compute_posterior_hypers(
        AbstractLikelihood& like,
        AbstractPriorModel& prior) override {
     // Likelihood and Prior downcast
     auto& likecast = downcast_likelihood(like);
     auto& priorcast = downcast_prior(prior);

     // Getting required quantities from likelihood and prior
     int card = likecast.get_card();
     double data_sum = likecast.get_data_sum();
     double ndata = likecast.get_ndata();
     double shape = priorcast.get_shape();
     auto hypers = priorcast.get_hypers();

     // No update possible
     if (card == 0) {
       return priorcast.get_hypers_proto();
     }
     // Compute posterior hyperparameters
     double rate_alpha_new = hypers.rate_alpha + shape * ndata;
     double rate_beta_new = hypers.rate_beta + data_sum;

     // Proto conversion
     ProtoHypers out;
     out.mutable_general_state()->
        mutable_data()->Add(rate_alpha_new);
     out.mutable_general_state()->
        mutable_data()->Add(rate_beta_new);
     return std::make_shared<ProtoHypers>(out);
   }
};
```

Note that implementing this new model has required only less than 130 lines of code. In particular, the coding effort could be substantially reduced by using, for instance, the `RandomWalkUpdater` instead of writing a custom `GammaGammaUpdater`.

13.8 Summary and Future Developments

In this chapter, we have presented **BayesMix**, a C++ library for posterior inference in Bayesian (nonparametric) mixture models. Compared to previously available software, our library features greater flexibility and extensibility, as shown by the modularity of our code, which makes it easy to extend our library to other mixture models. Therefore, **BayesMix** provides an ideal software ecosystem for computer scientists, statisticians and practitioners who need to consider complex models. As shown by the examples, our library compares favourably to the competitor package in terms of computational efficiency and of overall quality of the output MCMC samples.

The main limitation of **BayesMix** is also its point of strength, that is being a C++ library. As such, C++ programmers can benefit from the rich language and the efficiency of the C++ code to easily extend our library to their needs. However, knowledge of C++ might represent a barrier for new users.

To this end, we are currently developing the Python package **pybmix** (https://github.com/bayesmix-dev/pybmix), whose ultimate goal will be to allow the same degree of extensibility without knowledge of C++; users will be able to extend our library writing code solely in Python. Of course, this causes a loss in efficiency, since Python is slower than C++ and there issubstantial overhead in calling Python code from C++. However, compared to pure Python implementations, we expect our approach to be faster in terms of both runtime and development time (i.e., the time required to code an MCMC algorithm). We could certainly achieve the same goal within an R package, but at the moment this is not being considered.

The latest version of our library can be found at the official Github repository at https://github.com/bayesmix-dev/bayesmix. At the moment, our project has 14 contributors. Any interested user or developer can easily get in touch with us through our Github repository by opening an issue or requesting new features. We welcome any contribution to **BayesMix** and the Python package **pybmix**. Moreover, we would be happy to provide support to developers aiming at building an R package interface.

Acknowledgements

We thank all the developers who contributed to the **BayesMix** library and, in particular: Matteo Bollettino, Giacomo De Carlo, Alessandro Carminati, Madhurii Gatto, Taguhi Mesropyan, Vittorio Nardi, Enrico Paglia, Giovanni Battista Pollam, Pasquale Scaramuzzino. We are also grateful to Elena Zazzetti, who started working on this library when it was in an embryonal stage.

Bibliography

Agueh, M. and G. Carlier (2011). Barycenters in the Wasserstein space. *SIAM J. Math. Anal. 43*(2), 904–924.

Aitchison, J. (1986). *The Statistical Analysis of Compositional Data.* Chapman and Hall, London.

Aitchison, J. and S. M. Shen (1980). Logistic-normal distributions: Some properties and uses. *Biometrika 67*(2), 261–272.

Ali, M., B. Wainwright, A. Petersen, G. B. Jonnadula, M. Desai, H. L. Rao, M. Srinivas, S. R. Jammalamadaka, S. Senthil, S. Pyne, et al. (2021). Circular functional analysis of oct data for precise identification of structural phenotypes in the eye. *Scientific reports 11*(1), 1–13.

Álvarez-Esteban, P. C., E. del Barrio, J. A. Cuesta-Albertos, and C. Matrán (2018). Wide consensus aggregation in the Wasserstein space. Application to location-scatter families. *Bernoulli 24*(4A), 3147 – 3179.

Ambrosio, L., N. Gigli, and G. Savaré (2008). *Gradient flows: in metric spaces and in the space of probability measures.* Springer Science & Business Media.

Andrieu, C. and J. Thoms (2008). A tutorial on adaptive mcmc. *Statistics and computing 18*(4), 343–373.

Anevski, D., O. Hössjer, et al. (2006). A general asymptotic scheme for inference under order restrictions. *The Annals of Statistics 34*(4), 1874–1930.

Anevski, D. and P. Soulier (2011). Monotone spectral density estimation. *The Annals of Statistics 39*(1), 418–438.

Antoniak, C. E. (1974). Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, 1152–1174.

Aragam, B., C. Dan, E. P. Xing, and P. Ravikumar (2020). Identifiability of nonparametric mixture models and bayes optimal clustering. *The Annals of Statistics 48*(4), 2277–2302.

Arbel, J., E. Barrios, G. Kon-Kam-King, A. Lijoi, L. E. Nieto-Barajas, and I. Prünster (2020). *BNPdensity: Ferguson-Klass Type Algorithm for Posterior Normalized Random Measures.*

Argiento, R., I. Bianchini, and A. Guglielmi (2016). Posterior sampling from ε-approximation of normalized completely random measure mixtures. *Electronic Journal of Statistics 10*(2), 3516–3547.

Argiento, R., A. Cremaschi, and M. Vannucci (2019). Hierarchical normalized completely random measures to cluster grouped data. *Journal of the American Statistical Association 0*(0), 1–26.

Argiento, R. and M. De Iorio (2022). Is infinity that far? a bayesian nonparametric perspective of finite mixture models. *The Annals of Statistics - Future papers*.

Arminger, G. and B. O. Muthén (1998). A Bayesian approach to nonlinear latent variable models using the Gibbs sampler and the Metropolis-Hastings algorithm. *Psychometrika 63*(3), 271–300.

Arnold, B. C. and R. A. Groeneveld (1995). Measuring skewness with respect to the mode. *The American Statistician 49*(1), 34–38.

Aubin, J.-P. and H. Frankowska (2009). *Set-valued analysis.* Springer Science & Business Media.

Ayer, M., H. D. Brunk, G. M. Ewing, W. T. Reid, and E. Silverman (1955). An empirical distribution function for sampling with incomplete information. *The Annals of Mathematical Statistics 26*(4), 641–647.

Bacallado, S., M. Battiston, S. Favaro, and L. Trippa (2017). Sufficientness Postulates for Gibbs-Type Priors and Hierarchical Generalizations. *Statistical Science 32*(4), 487 – 500.

Baccelli, F., B. Błaszczyszyn, and M. Karray (2020). Random measures, point processes, and stochastic geometry. *HAL preprint available at https://hal.inria.fr/hal-02460214/*.

Banerjee, M., R. Chakraborty, E. Ofori, D. Vaillancourt, and B. C. Vemuri (2015). Nonlinear regression on riemannian manifolds and its applications to neuro-image analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 719–727. Springer.

Banerjee, S. (2016). Spatial data analysis. *Annual review of public health 37*, 47–60.

Banerjee, S., B. P. Carlin, and A. E. Gelfand (2014). *Hierarchical modeling and analysis for spatial data.* CRC press.

Bardenet, R. and M. Titsias (2015). Inference for determinantal point processes without spectral knowledge. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 28*, pp. 3393–3401. Curran Associates, Inc.

Barrientos, A. F., A. Jara, F. A. Quintana, et al. (2012). On the support of maceachern's dependent dirichlet processes and extensions. *Bayesian Analysis 7*(2), 277–310.

Barrios, E., A. Lijoi, L. E. Nieto-Barajas, and I. Prünster (2013). Modeling with normalized random measure mixture models. *Statistical Science 28*(3), 313 – 334.

Bassetti, F., A. Bodini, and E. Regazzini (2006). On minimum kantorovich distance estimators. *Statistics & probability letters 76*(12), 1298–1302.

Bassetti, F., R. Casarin, and L. Rossini (2020). Hierarchical species sampling models. *Bayesian Analysis 15*(3), 809–838.

Beraha, M., R. Argiento, J. Møller, and A. Guglielmi (2022). Mcmc computations for bayesian mixture models using repulsive point processes. *Journal of Computational and Graphical Statistics 0*(0), 1–14.

Beraha, M. and J. E. Griffin (2022). Normalized latent measure factor models. *ArXiv preprint, arXiv:2205.15654*.

Beraha, M. and A. Guglielmi (2019). Discussion on 'latent nested nonparametric priors' by camerlenghi, dunson, lijoi, prünster and rodriguez. *Bayesian Analysis 14*(4), 1326–1332.

Beraha, M., A. Guglielmi, and F. A. Quintana (2021). The semi-hierarchical Dirichlet Process and its application to clustering homogeneous distributions. *Bayesian Analysis 16*(4), 1187–1219.

Beraha, M., A. Guglielmi, F. A. Quintana, M. de Iorio, J. G. Eriksson, and F. Yap (2022). Bayesian nonparametric vector autoregressive models via a logit stick-breaking prior: an application to child obesity. *arXiv preprint, arXiv:2203.12280*.

Beraha, M., M. Pegoraro, R. Peli, and A. Guglielmi (2021). Spatially dependent mixture models via the logistic multivariate CAR prior. *Spatial Statistics 46*, 100548.

Berezin, S. and A. Miftakhov (2019). On barycenters of probability measures. *arXiv preprint arXiv:1911.07680*.

Bernton, E., P. E. Jacob, M. Gerber, and C. P. Robert (2019). On parameter estimation with the wasserstein distance. *Information and Inference: A Journal of the IMA 8*(4), 657–676.

Bernton, E., P. E. Jacob, M. Gerber, C. P. Robert, et al. (2019). Approximate bayesian computation with the wasserstein distance. *Journal of the Royal Statistical Society Series B 81*(2), 235–269.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological) 36*(2), 192–225.

Best, M. J. and N. Chakravarti (1990). Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming 47*(1-3), 425–439.

Bhadra, A., J. Datta, N. G. Polson, B. Willard, et al. (2019). Lasso meets horseshoe: A survey. *Statistical Science 34*(3), 405–427.

Bhattacharya, A. and D. Dunson (2012). Nonparametric bayes classification and hypothesis testing on manifolds. *Journal of multivariate analysis 111*, 1–19.

Bhattacharya, A. and D. B. Dunson (2011). Sparse Bayesian infinite factor models. *Biometrika 98*, 291–306.

Bhattacharya, A., D. Pati, N. S. Pillai, and D. B. Dunson (2015). Dirichlet–laplace priors for optimal shrinkage. *Journal of the American Statistical Association 110*(512), 1479–1490.

Bhattacharya, R. N., L. Ellingson, X. Liu, V. Patrangenaru, and M. Crane (2012). Extrinsic analysis on manifolds is computationally faster than intrinsic analysis with applications to quality control by machine vision. *Applied Stochastic Models in Business and Industry 28*(3), 222–235.

Bianchini, I., A. Guglielmi, and F. A. Quintana (2020). Determinantal point process mixtures via spectral density approach. *Bayesian Analysis 15*, 187–214.

Bigot, J., R. Gouet, T. Klein, A. López, et al. (2017). Geodesic PCA in the Wasserstein space by convex PCA. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, Volume 53, pp. 1–26. Institut Henri Poincaré.

Billio, M., R. Casarin, and L. Rossini (2019). Bayesian nonparametric sparse var models. *Journal of Econometrics 212*(1), 97–115.

Binder, D. A. (1978). Bayesian cluster analysis. *Biometrika 65*(1), 31–38.

Birgin, E. G. and J. M. Martinez (2014). *Practical augmented Lagrangian methods for constrained optimization.* SIAM.

Biscio, C. A. N., F. Lavancier, et al. (2016). Quantifying repulsiveness of determinantal point processes. *Bernoulli 22*(4), 2001–2028.

Blackwell, D. and J. B. MacQueen (1973). Ferguson distributions via pólya urn schemes. *The Annals of Statistics 1*(2), 353–355.

Blei, D. M. and M. I. Jordan (2006). Variational inference for dirichlet process mixtures. *Bayesian analysis 1*(1), 121–143.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent dirichlet allocation. *Journal of Machine Learning Research 3*(Jan), 993–1022.

Box, G. E. and D. R. Cox (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological) 26*(2), 211–243.

Brix, A. (1999). Generalized gamma measures and shot-noise Cox processes. *Adv. in Appl. Probab. 31*(4), 929–953.

Brunsdon, C., S. Fotheringham, and M. Charlton (1998). Geographically weighted regression. *Journal of the Royal Statistical Society: Series D (The Statistician) 47*(3), 431–443.

Cai, D., T. Campbell, and T. Broderick (2021). Finite mixture models do not reliably learn the number of components. In *International Conference on Machine Learning*, pp. 1158–1169. PMLR.

Cai, T. T. and P. Hall (2006). Prediction in functional linear regression. *The Annals of Statistics 34*, 2159–2179.

Camerlenghi, F., D. B. Dunson, A. Lijoi, I. Prünster, and A. Rodriguez (2019, 12). Latent nested nonparametric priors (with discussion). *Bayesian Anal. 14*(4), 1303–1356.

Camerlenghi, F., A. Lijoi, P. Orbanz, and I. Prünster (2019). Distribution theory for hierarchical processes. *The Annals of Statistics 47*(1), 67–92.

Campbell, T., J. H. Huggins, J. P. How, and T. Broderick (2019). Truncated random measures. *Bernoulli 25*, 1256–1288.

Campbell, T., S. Syed, C.-Y. Yang, M. I. Jordan, and T. Broderick (2019). Local exchangeability. *arXiv preprint arXiv:1906.09507*.

Canale, A., R. Corradin, and B. Nipoti (2019). Importance conditional sampling for pitman-yor mixtures. *arXiv preprint arXiv:1906.08147*.

Canova, F. and M. Ciccarelli (2004). Forecasting and turning point predictions in a bayesian panel var model. *Journal of Econometrics 120*(2), 327–359.

Cao, J., L. Mo, Y. Zhang, K. Jia, C. Shen, and M. Tan (2019). Multi-marginal Wasserstein GAN. In *Advances in Neural Information Processing Systems*, pp. 1776–1786.

Carlier, G., A. Oberman, and E. Oudet (2015). Numerical methods for matching for teams and wasserstein barycenters. *ESAIM: Mathematical Modelling and Numerical Analysis 49*(6), 1621–1642.

Carlin, B. P. and S. Chib (1995). Bayesian model choice via markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Methodological) 57*(3), 473–484.

Caron, F. and E. B. Fox (2017). Sparse graphs using exchangeable random measures. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 79*(5), 1295–1366.

Carpenter, B., A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell (2017). Stan: A probabilistic programming language. *Journal of statistical software 76*(1).

Carvalho, C. M., J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, and M. West (2008). High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association 103*(484), 1438–1456.

Catalano, M., A. Lijoi, and I. Prünster (2021). Measuring dependence in the wasserstein distance for bayesian nonparametric models. *The Annals of Statistics forthcoming.*

Catalano, P., N. Drago, and S. Amini (1995). Factors affecting fetal growth and body composition. *Am J Obstet Gynecol. 172*(5), 1459–63.

Cazelles, E., V. Seguy, J. Bigot, M. Cuturi, and N. Papadakis (2018). Geodesic PCA versus log-PCA of histograms in the Wasserstein space. *SIAM Journal on Scientific Computing 40*(2), B429–B456.

CDS (2018). Centers for disease control and prevention - behavior, environment, and genetic factors all have a role in causing people to be overweight and obese. Accessed: 19-01-2018.

Celeux, G., S. Frühwirth-Schnatter, and C. P. Robert (2019). Model selection for mixture models-perspectives and strategies. In *Handbook of mixture analysis*, pp. 117–154. Chapman & Hall/CRC.

Celeux, G., K. Kamary, G. Malsiner-Walli, J.-M. Marin, and C. P. Robert (2019). Computational solutions for bayesian inference in mixture models. *Handbook of Mixture Analysis*, 73–96.

Chandra, N. K., A. Canale, and D. B. Dunson (2020). Escaping the curse of dimensionality in bayesian model based clustering. *arXiv preprint arXiv:2006.02700*.

Chatterjee, D., T. Maitra, and S. Bhattacharya (2020). A short note on almost sure convergence of bayes factors in the general set-up. *The American Statistician 74*(1), 17–20.

Chen, Y. and T. E. Hanson (2014). Bayesian nonparametric k-sample tests for censored and uncensored data. *Computational Statistics and Data Analysis 71*, 335–346.

Chen, Y., Z. Lin, and H.-G. Müller (2021). Wasserstein regression*. *Journal of the American Statistical Association 0*(ja), 1–40.

Chib, S. and T. A. Kuffner (2016). Bayes factor consistency. *arXiv preprint arXiv:1607.00292*.

Chung, Y. and D. B. Dunson (2009). Nonparametric bayes conditional distribution modeling with variable selection. *Journal of the American Statistical Association 104*(488), 1646–1660.

Cifarelli, D. and E. Regazzini (1978). Problemi statistici non parametrici in condizioni di scambiabilita parziale e impiego di medie associative. Technical report, Tech. rep., Quaderni Istituto Matematica Finanziaria dell'Universita di Torino.

Clayton, D. and J. Kaldor (1987). Empirical bayes estimates of age-standardized relative risks for use in disease mapping. *Biometrics*, 671–681.

Coeurjolly, J.-F., J. Møller, and R. Waagepetersen (2017). A tutorial on palm distributions for spatial point processes. *International Statistical Review 85*(3), 404–420.

Collins, L. M. and S. T. Lanza (2009). *Latent Class and Latent Transition Analysis: With Applications in the Social, Behavioral, and Health Sciences.* John Wiley & Sons, New York.

Conti, G., S. Frühwirth-Schnatter, J. J. Heckman, and R. Piatek (2014). Bayesian exploratory factor analysis. *J. Econom. 183*, 31–57.

Coplien, J. O. (1995, feb). Curiously recurring template patterns. *C++ Rep. 7*(2), 24–27.

Corradin, R., A. Canale, and B. Nipoti (2020). *BNPmix: Bayesian Nonparametric Mixture Models.* R package version 0.2.7.

Cox, D. R. (1955, July). Some statistical methods connected with series of events. *Journal of the Royal Statistical Society: Series B (Methodological) 17*(2), 129–157.

Cremaschi, A., M. De Iorio, N. Kothandaraman, F. Yap, M. T. Tint, and J. Eriksson (2021). Integrating metabolic networks and growth biomarkers to unveil potential mechanisms of obesity. *arXiv preprint arXiv:2111.06212*.

Cressie, N. (1992). Statistics for spatial data. *Terra Nova 4*(5), 613–617.

Cressie, N. (1993). *Statistics for spatial data.* Wiley.

Cressie, N. and C. K. Wikle (2015). *Statistics for spatio-temporal data.* John Wiley & Sons.

Cuturi, M. (2013). Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems*, pp. 2292–2300.

Cuturi, M. and A. Doucet (2014). Fast Computation of Wasserstein Barycenters. In *International Conference on Machine Learning*, pp. 685–693.

Cuturi, M., L. Meng-Papaxanthos, Y. Tian, C. Bunne, G. Davis, and O. Teboul (2022). Optimal transport tools (ott): A jax toolbox for all things wasserstein. *arXiv preprint arXiv:2201.12324*.

Cuturi, M., O. Teboul, and J.-P. Vert (2019). Differentiable Ranking and Sorting using Optimal Transport. In *Advances in Neural Information Processing Systems*, pp. 6861–6871.

Daley, D. J. and D. Vere-Jones (2003). *An introduction to the theory of point processes. Vol. I* (Second ed.). Probability and its Applications (New York). New York: Springer-Verlag. Elementary theory and methods.

Daley, D. J. and D. Vere-Jones (2008). *An introduction to the theory of point processes. Vol. II* (Second ed.). Probability and its Applications (New York). New York: Springer. General theory and structure.

Daniels, M. J. and M. Pourahmadi (2002). Bayesian analysis of covariance matrices and dynamic models for longitudinal data. *Biometrika 89*(3), 553–566.

Das, P. and S. Ghosal (2017). Bayesian quantile regression using random B-spline series prior. *Computational Statistics & Data Analysis 109*, 121–143.

Dauxois, J., A. Pousse, and Y. Romain (1982). Asymptotic Theory for the Principal Component Analysis of a Vector Random Function: Some Applications to Statistical Inference. *Journal of Multivariate Analysis 12*, 136–154.

De Blasi, P., S. Favaro, A. Lijoi, R. H. Mena, I. Prünster, and M. Ruggiero (2013). Are gibbs-type priors the most natural generalization of the dirichlet process? *IEEE transactions on pattern analysis and machine intelligence 37*(2), 212–229.

De Blasi, P., A. Lijoi, and I. Prünster (2013). An asymptotic analysis of a class of discrete nonparametric priors. *Statistica Sinica*, 1299–1321.

De Boor, C. and J. W. Daniel (1974). Splines with Nonnegative B-spline Coefficients. *Mathematics of computation 28*(126), 565–568.

de Finetti, B. (1937). La prévision : ses lois logiques, ses sources subjectives. *Ann. Inst. H. Poincaré 7*(1), 1–68.

de Finetti, B. (1938). Sur la condition de “equivalence partielle”, colloque consacréa la théorie des probabilités. *VI, Université de Geneve, Hermann et C. ie, Paris*.

De Iorio, M., P. Müller, G. L. Rosner, and S. N. MacEachern (2004). An anova model for dependent random measures. *Journal of the American Statistical Association 99*(465), 205–215.

de Valpine, P., D. Turek, C. J. Paciorek, C. Anderson-Bergman, D. T. Lang, and R. Bodik (2017). Programming with models: Writing statistical algorithms for general model structures with nimble. *Journal of Computational and Graphical Statistics 26*(2), 403–413.

Delicado, P. (2011, 01). Dimensionality reduction when data are density functions. *Computational Statistics & Data Analysis 55*, 401–420.

Dellaportas, P., J. J. Forster, and I. Ntzoufras (2002, Jan). On bayesian model and variable selection using mcmc. *Statistics and Computing 12*(1), 27–36.

Dellaportas, P. and I. Papageorgiou (2006). Multivariate mixtures of normals with unknown number of components. *Statistics and Computing 16*(1), 57–68.

Delon, J., J. Salomon, and A. Sobolevski (2010). Fast transport optimization for monge costs on the circle. *SIAM Journal on Applied Mathematics 70*(7), 2239–2258.

Després, J.-P., I. Lemieux, J. Bergeron, P. Pibarot, P. Mathieu, E. Larose, J. Rodés-Cabau, O. F. Bertrand, and P. Poirier (2008). Abdominal obesity and the metabolic syndrome: Contribution to global cardiometabolic risk. *Arteriosclerosis, Thrombosis, and Vascular Biology 28*(6), 1039–1049.

Deutsch, F. (2012). *Best Approximation in Inner-Product Spaces.* Springer Science & Business Media.

Duan, J. A., M. Guindani, and A. E. Gelfand (2007). Generalized spatial dirichlet process models. *Biometrika 94*(4), 809–825.

Duan, L. L. and D. B. Dunson (2021). Bayesian distance clustering. *Journal of Machine Learning Research 22*(224), 1–27.

Dunson, D. B. and J.-H. Park (2008). Kernel stick-breaking processes. *Biometrika 95*(2), 307–323.

Dunstan, P. K., S. D. Foster, F. K. Hui, and D. I. Warton (2013). Finite mixture of regression modeling for high-dimensional count and biomass data in ecology. *Journal of agricultural, biological, and environmental statistics 18*(3), 357–375.

Dykstra, R., T. Robertson, and F. T. Wright (2012). *Advances in Order Restricted Statistical Inference: Proceedings of the Symposium on Order Restricted Statistical Inference Held in Iowa City, Iowa, September 11–13, 1985*, Volume 37. Springer Science & Business Media.

Eaton, M. L. (1983). Multivariate statistics: a vector space approach. *John Wiley & Sons, Inc., New York.*

Egozcue, J. J., J. L. Diaz-Barrero, and V. Pawlowsky-Glahn (2006). Hilbert Space of Probability Density Functions Based on Aitchison Geometry. *Acta Mathematica Sinica 22*(4), 1175–1182.

Elliott, L. T., M. D. Iorio, S. Favaro, K. Adhikari, and Y. W. Teh (2019). Modeling Population Structure Under Hierarchical Dirichlet Processes. *Bayesian Anal. 14*(2), 313 – 339.

Escobar, M. D. and M. West (1995). Bayesian density estimation and inference using mixtures. *Journal of the american statistical association 90*(430), 577–588.

Ewens, W. J. (1972). The sampling theory of selectively neutral alleles. *Theoretical population biology 3*(1), 87–112.

Favaro, S., G. Hadjicharalambous, and I. Prünster (2011). On a class of distributions on the simplex. *Journal of Statistical Planning and Inference 141*(9), 2987–3004.

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

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

Ferguson, T. S. (1983). Bayesian density estimation by mixtures of normal distributions. In *Recent advances in statistics*, pp. 287–302. Elsevier.

Fields, D., S. Krishnan, and A. Wisniewski (2009). Sex differences in body composition early in life. *Gend Med. 6*(2), 369–75.

Flaming, D. and M. Matsunaga (2008). Concentrated poverty in Los Angeles (february 9, 2008). *Economic Roundtable Research Report, February 2008, Available at SSRN: https://ssrn.com/abstract=2772191*.

Fletcher, P. (2013, 11). Geodesic Regression and the Theory of Least Squares on Riemannian Manifolds. *International Journal of Computer Vision 105*.

Fletcher, P. T., C. Lu, S. M. Pizer, and S. Joshi (2004). Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging 23*(8), 995–1005.

Fox, C. S., J. M. Massaro, U. Hoffmann, K. M. Pou, P. Maurovich-Horvat, C.-Y. Liu, R. S. Vasan, J. M. Murabito, J. B. Meigs, L. A. Cupples, R. B. D'Agostino, and C. J. O'Donnell (2007). Abdominal visceral and subcutaneous adipose tissue compartments. *Circulation 116*(1), 39–48.

Fraley, C. and A. E. Raftery (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification 24*(2), 155–181.

França, G., A. Barp, M. Girolami, and M. I. Jordan (2021). Optimization on manifolds: A symplectic approach.

Frühwirth-Schnatter, S. (2006). *Finite mixture and Markov switching models*. Springer Science & Business Media.

Fruhwirth-Schnatter, S., G. Celeux, and C. P. Robert (2019). *Handbook of Mixture Analysis*. Chapman and Hall/CRC, New York.

Frühwirth-Schnatter, S. and G. Malsiner-Walli (2019). From here to infinity: sparse finite versus dirichlet process mixtures in model-based clustering. *Advances in data analysis and classification 13*(1), 33–64.

Frühwirth-Schnatter, S., G. Malsiner-Walli, and B. Grün (2021). Generalized mixtures of finite mixtures and telescoping sampling. *Bayesian Analysis 16*(4), 1279–1307.

Fúquene, J., M. Steel, and D. Rossell (2019). On choosing mixture components via non-local priors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 81*(5), 809–837.

Gao, F., K. I. Zheng, X.-B. Wang, Q.-F. Sun, K.-H. Pan, T.-Y. Wang, Y.-P. Chen, G. Targher, C. D. Byrne, J. George, et al. (2020). Obesity is a risk factor for greater covid-19 severity. *Diabetes care 43*(7), e72–e74.

Garcia-Ayllon, S. (2018). Urban transformations as an indicator of unsustainability in the p2p mass tourism phenomenon: The airbnb case in spain through three case studies. *Sustainability 10*(8), 2933.

Geisser, S. and W. F. Eddy (1979). A predictive approach to model selection. *Journal of the American Statistical Association 74*(365), 153–160.

Gelfand, A. E., A. Kottas, and S. N. MacEachern (2005). Bayesian nonparametric spatial modeling with dirichlet process mixing. *Journal of the American Statistical Association 100*(471), 1021–1035.

Gelfand, A. E. and P. Vounatsou (2003). Proper multivariate conditional autoregressive models for spatial data analysis. *Biostatistics 4*(1), 11–15.

Gelman, A., X.-L. Meng, and H. Stern (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica sinica*, 733–760.

Geweke, J. and G. Zhou (1996). Measuring the pricing error of the arbitrage pricing theory. *The review of financial studies 9*(2), 557–587.

Geweke, J. and G. Zhou (2015, 06). Measuring the Pricing Error of the Arbitrage Pricing Theory. *Rev. Financ. Stud. 9*(2), 557–587.

Geweke, J. F. and K. J. Singleton (1980). Interpreting the likelihood ratio statistic in factor models when sample size is small. *J. Am. Stat. Assoc. 75*(369), 133–137.

Geyer, C. J. and J. Møller (1994). Simulation procedures and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics 21*, 359–373.

Ghodrati, L. and V. M. Panaretos (2021). Distribution-on-distribution regression via optimal transport maps. *arXiv preprint arXiv:2104.09418*.

Ghosal, S., J. K. Ghosh, R. Ramamoorthi, et al. (1999). Posterior consistency of dirichlet mixtures in density estimation. *Ann. Statist 27*(1), 143–158.

Ghosal, S., J. Lember, and A. Van Der Vaart (2008). Nonparametric bayesian model selection and averaging. *Electronic Journal of Statistics 2*, 63–89.

Ghosal, S. and A. Van der Vaart (2017). *Fundamentals of nonparametric Bayesian inference*, Volume 44. Cambridge University Press.

Gibbs, A. L. and F. E. Su (2002). On choosing and bounding probability metrics. *International statistical review 70*(3), 419–435.

Gigli, N. (2011). On the inverse implication of brenier-mccann theorems and the structure of (p 2 (m), w 2). *Methods and Applications of Analysis 18*(2), 127–158.

Gnedin, A. and J. Pitman (2005). Exchangeable Gibbs partitions and Stirling triangles. *Zap. Nauchn. Sem. S.-Peterburg. Otdel. Mat. Inst. Steklov. (POMI) 325*(Teor. Predst. Din. Sist. Komb. i Algoritm. Metody. 12), 83–102, 244–245.

Godfrey, K. M., G. Haugen, T. Kiserud, H. M. Inskip, C. Cooper, N. C. W. Harvey, S. R. Crozier, S. M. Robinson, L. Davies, the Southampton Women's Survey Study Group, and M. A. Hanson (2012, 08). Fetal liver blood flow distribution: Role in human developmental strategy to prioritize fat deposition versus brain development. *PLOS ONE 7*(8), 1–7.

Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika 82*(4), 711–732.

Green, P. J. (2010). Trans-dimensional markov chain monte carlo. In P. J. Green, N. L. Hjort, and S. Richardson (Eds.), *Highly Structured Stochastic Systems*, pp. 179–198. Oxford University Press, Oxford U.K.

Griewank, A. et al. (1989). On automatic differentiation. *Mathematical Programming: recent developments and applications 6*(6), 83–107.

Griffin, J. E., M. Kolossiatis, and M. F. J. Steel (2013). Comparing distributions by using dependent normalized random-measure mixtures. *J. R. Statist. Soc. B 75*(3), 499–529.

Griffin, J. E. and F. Leisen (2017). Compound random measures and their use in Bayesian non-parametrics. *J. R. Statist. Soc. B 79*(2), 525–545.

Griffin, J. E. and S. G. Walker (2011). Posterior simulation of normalized random measure mixtures. *Journal of Computational and Graphical Statistics 20*(1), 241–259.

Griffith, D., Y. Chun, and B. Li (2019). *Spatial regression analysis using eigenvector spatial filtering.* Academic Press.

Guha, A., N. Ho, and X. Nguyen (2021). On posterior contraction of parameters and interpretability in bayesian mixture modeling. *Bernoulli 27*(4), 2159–2188.

Gutiérrez, L., A. F. Barrientos, J. González, and D. Taylor-Rodriguez (2019, 06). A bayesian nonparametric multiple testing procedure for comparing several treatments against a control. *Bayesian Anal. 14*(2), 649–675.

Haario, H., E. Saksman, and J. Tamminen (2001). An adaptive Metropolis algorithm. *Bernoulli 7*(2), 223 – 242.

Hales, C. M., C. D. Fryar, M. D. Carroll, D. S. Freedman, and C. L. Ogden (2018). Trends in obesity and severe obesity prevalence in us youth and adults by sex and age, 2007-2008 to 2015-2016. *Jama 319*(16), 1723–1725.

Holmes, C. C., F. Caron, J. E. Griffin, and D. A. Stephens (2015). Two-sample bayesian nonparametric hypothesis testing. *Bayesian Analysis 10*(2), 297–320.

Holmes, C. C. and L. Held (2006). Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian analysis 1*(1), 145–168.

Hough, J. B., M. Krishnapur, Y. Peres, and B. Viràg (2006). Determinantal processes and independence. *Probability Surveys 3*, 206–229.

Hough, J. B., M. Krishnapur, Y. Peres, and B. Virág (2009). *Zeros of Gaussian Analytic Functions and Determinantal Point Processes.* Providence: American Mathematical Society.

Hron, K., A. Menafoglio, M. Templ, K. Hrůzová, and P. Filzmoser (2014, 07). Simplicial principal component analysis for density functions in Bayes spaces. *Computational Statistics & Data Analysis 94*, 330–350.

Hubert, L. and P. Arabie (1985). Comparing partitions. *Journal of classification 2*(1), 193–218.

Huckemann, S., T. Hotzand, and A. Munk (2010). Intrinsic shape analysis: Geodesic PCA for Riemannian manifolds modulo isometric lie group actions. *Statistica Sinica 20*, 1–58.

Huckemann, S. F. and B. Eltzner (2018). Backward nested descriptors asymptotics with inference on stem cell differentiation. *The Annals of Statistics 46*(5), 1994–2019.

Hughes, M. C. and E. B. Sudderth (2014). bnpy: Reliable and scalable variational inference for bayesian nonparametric models. *Probabilistic Programming Workshop at NIPS*.

Hyvärinen, A. (2013). Independent component analysis: recent advances. *Philos. Trans. Royal Soc. A 371*(1984), 20110534.

Ishwaran, H. and L. F. James (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association 96*(453), 161–173.

Ishwaran, H. and L. F. James (2002). Approximate dirichlet process computing in finite normal mixtures: smoothing and prior information. *Journal of Computational and Graphical statistics 11*(3), 508–532.

Jain, S. and R. M. Neal (2004). A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of computational and Graphical Statistics 13*(1), 158–182.

Jakob, W., J. Rhinelander, and D. Moldovan (2017). pybind11 – seamless operability between c++11 and python. https://github.com/pybind/pybind11.

James, L. F., A. Lijoi, and I. Prünster (2009). Posterior analysis for normalized random measures with independent increments. *Scandinavian Journal of Statistics 36*(1), 76–97.

Janati, H., M. Cuturi, and A. Gramfort (2020). Debiased sinkhorn barycenters. In *International Conference on Machine Learning*, pp. 4692–4701. PMLR.

Jara, A., T. Hanson, F. A. Quintana, P. Müller, and G. L. Rosner (2011). Dppackage: Bayesian semi- and nonparametric modeling in r. *Journal of Statistical Software 40*(5), 1–30.

Jo, S., J. Lee, P. Müller, F. A. Quintana, and L. Trippa (2017). Dependent species sampling models for spatial density estimation. *Bayesian Analysis 12*(2), 379–406.

Joshi, N., S. Kulkarni, C. Yajnik, C. Joglekar, S. Rao, K. Coyaji, L. H.G., S. Rege, and C. Fall (2005). Increasing maternal parity predicts neonatal adiposity: Pune Maternal Nutrition Study. *Am J Obstet Gynecol Sep;193*(3 Pt 1), 783–9.

Jung, S., I. L. Dryden, and J. S. Marron (2012). Analysis of principal nested spheres. *Biometrika 99*(3), 551–568.

Kaiser, M. S. and N. Cressie (2000). The construction of multivariate distributions from markov random fields. *Journal of Multivariate Analysis 73*(2), 199–220.

Kallenberg, O. (1984). An informal guide to the theory of conditioning in point processes. *International Statistical Review / Revue Internationale de Statistique 52*(2), 151–164.

Kallenberg, O. (2017). *Random measures, theory and applications*, Volume 1. Springer.

Kallenberg, O. ([2021] ©2021). *Foundations of modern probability*, Volume 99 of *Probability Theory and Stochastic Modelling.* Springer, Cham. Third edition.

Kalli, M. and J. E. Griffin (2018). Bayesian nonparametric vector autoregressive models. *Journal of econometrics 203*(2), 267–282.

Kalli, M., J. E. Griffin, and S. G. Walker (2011). Slice sampling mixture models. *Statistics & computing 21*(1), 93–105.

Keeler, H. P. and B. Błaszczyszyn (2014). Sinr in wireless networks and the two-parameter poisson-dirichlet process. *IEEE wireless communications letters 3*(5), 525–528.

Kendall, W. S. and J. Møller (2000). Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Advances in Applied Probability 32*, 844–865.

Kim, Y.-H. and B. Pass (2017). Wasserstein barycenters over Riemannian manifolds. *Adv. Math. 307*, 640–683.

Kingman, J. F. C. (1967). Completely random measures. *Pacific Journal of Mathematics 21*(1), 59 – 78.

Kingman, J. F. C. (1992). *Poisson processes*, Volume 3. Clarendon Press.

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

Kiselev, V. Y., T. S. Andrews, and M. Hemberg (2019). Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics 20*(5), 273–282.

Kleijn, B. J., A. W. van der Vaart, et al. (2006). Misspecification in infinite-dimensional bayesian statistics. *The Annals of Statistics 34*(2), 837–877.

Kneip, A. and K. J. Utikal (2001). Inference for Density Families Using Functional Principal Component Analysis. *Journal of the American Statistical Association 96*(454), 519–542.

Kriegel, H.-P., P. Kröger, and A. Zimek (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM transactions on knowledge discovery from data 3*(1), 1–58.

Kundu, S. and J. Lukemire (2021). Non-parametric bayesian vector autoregression using multi-subject data. *arXiv preprint arXiv:2111.08743*.

Lau, J. W. and P. J. Green (2007). Bayesian model-based clustering procedures. *Journal of Computational and Graphical Statistics 16*(3), 526–558.

Lavancier, F., J. Møller, and E. Rubak (2015). Determinantal point process models and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 77*(4), 853–877.

Le-Rademacher, J. and L. Billard (2017). Principal component analysis for histogram-valued data. *Advances in Data Analysis and Classification 11*(2), 327–351.

Lee, D. (2013a). CARBayes: An R package for Bayesian spatial modeling with conditional autoregressive priors. *Journal of Statistical Software 55*(13), 1–24.

Lee, D. and R. Mitchell (2012). Boundary detection in disease mapping studies. *Biostatistics 13*(3), 415–426.

Lee, J. M. (2013b). *Introduction to smooth manifolds* (Second ed.), Volume 218 of *Graduate Texts in Mathematics*. Springer, New York.

Legramanti, S., D. Durante, and D. B. Dunson (2020). Bayesian cumulative shrinkage for infinite factorizations. *Biometrika 107*, 745–752.

Leisen, F. and A. Lijoi (2011). Vectors of two-parameter poisson–dirichlet processes. *Journal of Multivariate Analysis 102*(3), 482–495.

Leisen, F., A. Lijoi, and D. Spanó (2013). A vector of dirichlet processes. *Electronic Journal of Statistics 7*, 62–90.

Leroux, B. G., X. Lei, and N. Breslow (2000). Estimation of disease rates in small areas: a new mixed model for spatial dependence. In *Statistical models in epidemiology, the environment, and clinical trials*, pp. 179–191. Springer.

Li, P., S. Banerjee, T. A. Hanson, and A. M. McBean (2015). Bayesian models for detecting difference boundaries in areal data. *Statistica Sinica 25*(1), 385.

Li, Y., X. Lin, and P. Müller (2010). Bayesian inference in semiparametric mixed models for longitudional data. *Biometrics 66*(1), 70–78.

Li, Y., J. Lord-Bessen, M. Shiyko, and R. Loeb (2018). Bayesian latent class analysis tutorial. *Multivariate Behavioral Research 53*(3), 430–451.

Liang, F., I. H. Jin, Q. Song, and J. S. Liu (2016). An adaptive exchange algorithm for sampling from distributions with intractable normalizing constants. *Journal of the American Statistical Association 111*(513), 377–393.

Lijoi, A., R. H. Mena, and I. Prünster (2005). Hierarchical mixture modeling with normalized inverse-gaussian priors. *Journal of the American Statistical Association 100*(472), 1278–1291.

Lijoi, A., R. H. Mena, and I. Prünster (2007). Controlling the reinforcement in bayesian non-parametric mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69*(4), 715–740.

Lijoi, A., B. Nipoti, and I. Prünster (2014a). Bayesian inference with dependent normalized completely random measures. *Bernoulli 20*(3), 1260–1291.

Lijoi, A., B. Nipoti, and I. Prünster (2014b). Dependent mixture models: clustering and borrowing information. *Computational Statistics & Data Analysis 71*, 417–433.

Lijoi, A. and I. Prünster (2010). Models beyond the dirichlet process. In N. Hjort, C. Holmes, P. Müller, and S. Walker (Eds.), *Bayesian Nonparametrics*, pp. 80–136. Cambridge University Press, Cambridge.

Lijoi, A., I. Prünster, and G. Rebaudo (2020). Flexible clustering via hidden hierarchical dirichlet priors. *Collegio Carlo Alberto Notebooks* (634).

Lijoi, A., I. Prünster, and T. Rigon (2020a). Finite-dimensional discrete random structures and bayesian clustering. *Preprint*.

Lijoi, A., I. Prünster, and T. Rigon (2020b). The pitman–yor multinomial process for mixture modeling. *Biometrika 107*(4), 891–906.

Lijoi, A., I. Prünster, and T. Rigon (2020c). Sampling hierarchies of discrete random structures. *Statistics and Computing 30*(6), 1591–1607.

Lijoi, A., I. Prünster, and S. G. Walker (2005). On consistency of nonparametric normal mixtures for bayesian density estimation. *Journal of the American Statistical Association 100*(472), 1292–1296.

Lijoi, A., I. Prünster, and S. G. Walker (2008). Investigating nonparametric priors with Gibbs structure. *Statist. Sinica 18*(4), 1653–1668.

Lin, Q., G. Rebaudo, and P. Mueller (2021). Separate exchangeability as modeling principle in bayesian nonparametrics. *arXiv preprint arXiv:2112.07755*.

Lo, A. Y. (1984). On a class of Bayesian nonparametric estimates. I. Density estimates. *Ann. Statist. 12*(1), 351–357.

Lü, H., J. Arbel, and F. Forbes (2020). Bayesian nonparametric priors for hidden markov random fields. *Statistics and Computing 30*(4), 1015–1035.

Lu, H., C. S. Reilly, S. Banerjee, and B. P. Carlin (2007). Bayesian areal wombling via adjacency modeling. *Environmental and ecological statistics 14*(4), 433–452.

Lucas, J., C. Carvalho, Q. Wang, A. Bild, J. R. Nevins, and M. West (2006). Sparse statistical modelling in gene expression genomics. *Bayesian inference for gene expression and proteomics 1*(1), 3.

Lyne, A.-M., M. Girolami, Y. Atchadé, H. Strathmann, D. Simpson, et al. (2015). On russian roulette estimates for Bayesian inference with doubly-intractable likelihoods. *Statistical Science 30*(4), 443–467.

Ma, L. and W. H. Wong (2011). Coupling optional pólya trees and the two sample problem. *Journal of the American Statistical Association 106*(496), 1553–1565.

Macchi, O. (1975). The coincidence approach to stochastic point processes. *Advances in Applied Probability 7*, 83–122.

MacEachern, S. N. (1999). Dependent nonparametric processes. In *ASA proceedings of the section on Bayesian statistical science*, Volume 1, pp. 50–55. Alexandria, Virginia. Virginia: American Statistical Association; 1999.

MacEachern, S. N. (2000). Dependent dirichlet processes. *Unpublished manuscript, Department of Statistics, The Ohio State University*, 1–40.

Malsiner-Walli, G., S. Frühwirth-Schnatter, and B. Grün (2016). Model-based clustering based on sparse finite gaussian mixtures. *Statistics and computing 26*(1), 303–324.

Malsiner-Walli, G., S. Frühwirth-Schnatter, and B. Grün (2017). Identifying mixtures of mixtures using bayesian estimation. *Journal of Computational and Graphical Statistics 26*(2), 285–295. PMID: 28626349.

Mandavilli, A. (2020, 08). Why does the coronavirus hit men harder? a new clue.

Mardia, K. (1988). Multi-dimensional multivariate gaussian markov random fields with application to image processing. *Journal of Multivariate Analysis 24*(2), 265–284.

McCann, R. J. (2001). Polar factorization of maps on Riemannian manifolds. *Geom. Funct. Anal. 11*(3), 589–608.

Mcvinish, R., J. Rousseau, and K. Mengersen (2009). Bayesian goodness of fit testing with mixtures of triangular distributions. *Scandinavian Journal of Statistics 36*(2), 337–354.

Menafoglio, A., A. Guadagnini, and P. Secchi (2014). A kriging approach based on Aitchison geometry for the characterization of particle-size curves in heterogeneous aquifers. *Stochastic Environmental Research and Risk Assessment 28*(7), 1835–1851.

Miller, J. W. and D. B. Dunson (2019). Robust bayesian inference via coarsening. *Journal of the American Statistical Association 114*(527), 1113–1125.

Miller, J. W. and M. T. Harrison (2014a). Inconsistency of pitman-yor process mixtures for the number of components. *Journal of Machine Learning Research 15*(96), 3333–3370.

Miller, J. W. and M. T. Harrison (2014b). Inconsistency of pitman-yor process mixtures for the number of components. *The Journal of Machine Learning Research 15*(1), 3333–3370.

Miller, J. W. and M. T. Harrison (2018). Mixture models with a prior on the number of components. *Journal of the American Statistical Association 113*(521), 340–356.

Misra, A. and L. Khurana (2011). Obesity-related non-communicable diseases: South asians vs white caucasians. *International journal of obesity 35*(2), 167–187.

Mitra, R. and P. Müller (2015). *Nonparametric Bayesian inference in biostatistics.* Springer.

Molitor, J., M. Papathomas, M. Jerrett, and S. Richardson (2010). Bayesian profile regression with an application to the national survey of children's health. *Biostatistics 11*(3), 484–498.

Møller, J. (2003, September). Shot noise cox processes. *Advances in Applied Probability 35*(3), 614–640.

Møller, J. and E. O'Reilly (2021). Couplings for determinantal point processes and their reduced palm distributions with a view to quantifying repulsiveness. *Advances in Applied Probability* (to appear). Available at arXiv:1806.07347.

Møller, J., A. N. Pettitt, R. Reeves, and K. K. Berthelsen (2006). An efficient markov chain monte carlo method for distributions with intractable normalising constants. *Biometrika 93*(2), 451–458.

Møller, J. and N. Vihrs (2022). Determinantal shot noise cox processes. *arXiv:2112.04204*.

Møller, J. and R. P. Waagepetersen (2004). *Statistical Inference and Simulation for Spatial Point Processes.* Chapman and Hall/CRC, Boca Raton.

Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*.

Montagna, S., S. T. Tokdar, B. Neelon, and D. B. Dunson (2012). Bayesian latent factor regression for functional and longitudinal data. *Biometrics 68*(4), 1064–1073.

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

Müller, P., F. Quintana, and G. Rosner (2004). A method for combining inference across related nonparametric Bayesian models. *J. R. Stat. Soc. B 66*(3), 735–749.

Müller, P., F. Quintana, and G. L. Rosner (2011). A product partition model with regression on covariates. *Journal of Computational and Graphical Statistics 20*(1), 260–278.

Müller, P., F. A. Quintana, A. Jara, and T. Hanson (2015). *Bayesian nonparametric data analysis*. Springer Series in Statistics. Springer, Cham.

Munkres, J. R. (2000). *Topology*. Prentice Hall, Inc., Upper Saddle River, NJ. Second edition of [MR0464128].

Murray, I., Z. Ghahramani, and D. J. C. MacKay (2006). Mcmc for doubly-intractable distributions. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'06, Arlington, Virginia, USA, pp. 359–366. AUAI Press.

Nagabhushan, P. and R. Pradeep Kumar (2007). Histogram PCA. In D. Liu, S. Fei, Z. Hou, H. Zhang, and C. Sun (Eds.), *Advances in Neural Networks – ISNN 2007*, Berlin, Heidelberg, pp. 1012–1021. Springer Berlin Heidelberg.

Natarajan, A., M. De Iorio, A. Heinecke, E. Mayer, and S. Glenn (2021). Cohesion and repulsion in bayesian distance clustering. *arXiv preprint arXiv:2107.05414*.

Neal, R. M. (2000). Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics 9*(2), 249–265.

Neal, R. M. (2003). Density modeling and clustering using dirichlet diffusion trees.

Neal, R. M. et al. (2011). Mcmc using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo 2*(11), 2.

Nguyen, T. D., J. H. Huggins, L. Masoero, L. Mackey, and T. Broderick (2020). Independent versus truncated finite approximations for Bayesian nonparametric inference. In *"I Can't Believe It's Not Better!" NeurIPS 2020 workshop.*

Nightingale, C. M., A. R. Rudnicka, C. G. Owen, D. G. Cook, and P. H. Whincup (2010, 11). Patterns of body size and adiposity among UK children of South Asian, black African–Caribbean and white European origin: Child Heart And health Study in England (CHASE Study). *International Journal of Epidemiology 40*(1), 33–44.

Nobile, A. (1994). *Bayesian analysis of finite mixture distributions*. Carnegie Mellon University.

Orbanz, P. and D. M. Roy (2014). Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE transactions on pattern analysis and machine intelligence 37*(2), 437–461.

Page, G. L. and F. A. Quintana (2016). Spatial product partition models. *Bayesian Analysis 11*(1), 265–298.

Panaretos, V. M. and Y. Zemel (2020). *An Invitation to Statistics in Wasserstein Space*. Springer Nature.

Papaspiliopoulos, O. and G. O. Roberts (2008). Retrospective markov chain monte carlo methods for dirichlet process hierarchical models. *Biometrika 95*(1), 169–186.

Papastamoulis, P. and I. Ntzoufras (2022, feb). On the identifiability of Bayesian factor analytic models. *Stat. Comput. 32*(2).

Park, J.-H. and D. B. Dunson (2010). Bayesian generalized product partition model. *Statistica Sinica*, 1203–1226.

Patrangenaru, V. and L. Ellingson (2015). *Nonparametric Statistics on Manifolds and Their Application to Object Data Analysis*. CRC Press.

Pegoraro, M. and M. Beraha (2022). Projected Statistical Methods for Distributional Data on the Real Line with the Wasserstein Metric. *Journal of Machine Learning Research 23*(37), 1–59.

Pennec, X. (2006, 07). Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *Journal of Mathematical Imaging and Vision 25*, 127–154.

Pennec, X. (2008). Statistical Computing on Manifolds: From Riemannian geometry to Computational Anatomy. In *LIX Fall Colloquium on Emerging Trends in Visual Computing*, pp. 347–386. Springer.

Pennec, X. (2018). Barycentric subspace analysis on manifolds. *The Annals of Statistics 46*(6A), 2711–2746.

Petersen, K. B. and M. S. Pedersen (2012, nov). The matrix cookbook. Version 2012/11/15.

Petralia, F., V. Rao, and D. B. Dunson (2012). Repulsive mixtures. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25*, pp. 1889–1897. Curran Associates, Inc.

Peyré, G., M. Cuturi, et al. (2019). Computational Optimal Transport: With Applications to Data Science. *Foundations and Trends in Machine Learning 11*(5-6), 355–607.

Pi-Sunyer, X. (2009). The medical risks of obesity. *Postgraduate medicine 121*(6), 21–33.

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability theory and related fields 102*(2), 145–158.

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.*, pp. 245–267. Inst. Math. Statist., Hayward, CA.

Pitman, J. (2006). *Combinatorial Stochastic Processes: Ecole d'Eté de Probabilités de Saint-Flour XXXII-2002*. Springer.

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

Plummer, M. (2003). Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing*, Number 125.10, pp. 1–10. Vienna, Austria.

Plummer, M. (2017, June). *JAGS Version 4.3.0 user manual.*

Poinas, A. and F. Lavancier (2021). Asymptotic approximation of the likelihood of stationary determinantal point processes. Technical report, available at arXiv:2103.02310.

Polson, N. G., J. G. Scott, and J. Windle (2013). Bayesian inference for logistic models using pólya-gamma latent variables. *Journal of the American Statistical Association 108*(504), 1339–1349.

Potra, F. A. and S. J. Wright (2000). Interior-point methods. *Journal of Computational and Applied Mathematics 124*(1-2), 281–302.

Potter, S., M. Del Negro, G. Topa, and W. Van der Klaauw (2017). The advantages of probabilistic survey questions. *Review of Economic Analysis 9*(1), 1–32.

Poworoznek, E., F. Ferrari, and D. Dunson (2021). Efficiently resolving rotational ambiguity in Bayesian matrix sampling with matching.

Prchal, L. and P. Sarda (2007). Spline estimator for functional linear regression with functional response. *Technical Report*.

Propp, J. G. and D. B. Wilson (1996). Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures & Algorithms 9*(1-2), 223–252.

Pya, N. and S. N. Wood (2015). Shape constrained additive models. *Statistics and Computing 25*(3), 543–559.

Pyrcz, M. J. and C. V. Deutsch (2014). *Geostatistical reservoir modeling.* Oxford university press.

Qasim, A., M. Turcotte, R. De Souza, M. Samaan, D. Champredon, J. Dushoff, J. Speakman, and D. Meyre (2018). On the origin of obesity: identifying the biological, environmental and cultural drivers of genetic risk among human populations. *Obesity reviews 19*(2), 121–149.

Qi, X. and H. Zhao (2011). Some theoretical properties of Silverman's method for smoothed functional principal component analysis. *Journal of Multivariate Analysis 102*, 741–767.

Quinlan, J. J., F. A. Quintana, and G. L. Page (2020). Parsimonious hierarchical modeling using repulsive distributions. *Test* (to appear).

Quintana, F. A. (1998). Nonparametric Bayesian Analysis for Assessing Homogeneity in k × l Contingency Tables with Fixed Right Margin Totals. *Journal of the American Statistical Association 93*(443), 1140–1149.

Quintana, F. A., W. O. Johnson, L. E. Waetjen, and E. B. Gold (2016). Bayesian nonparametric longitudinal data analysis. *Journal of the American Statistical Association 111*(515), 1168–1181.

Quintana, F. A., P. Mueller, A. Jara, and S. N. MacEachern (2022). The dependent dirichlet process and related models. *Statistical Science 37*(1), 24–41.

Quintana, F. A., P. Müller, A. Jara, and S. N. MacEachern (2022). The dependent Dirichlet process and related models. *Stat. Sci. 37*(1), 24–41.

Ramsay, J. O. (2004). Functional data analysis. *Encyclopedia of Statistical Sciences 4.*

Regazzini, E. (1991). Coherence, exchangeability and statistical models (de finetti's stance revisited). In *Atti del Convegno "Sviluppi metodologici nei diversi approcci all'inferenza statistica"*, pp. 101–137. Bologna: Pitagora.

Regazzini, E., A. Lijoi, and I. Prünster (2003). Distributional results for means of normalized random measures with independent increments. *Ann. Stat. 31*(2), 560 – 585.

Ren, L., L. Du, L. Carin, and D. Dunson (2011). Logistic stick-breaking process. *Journal of Machine Learning Research 12*(Jan), 203–239.

Richardson, S. and P. J. Green (1997). On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 59*(4), 731–792.

Rigon, T. and D. Durante (2021). Tractable bayesian density regression via logit stick-breaking priors. *Journal of Statistical Planning and Inference 211*, 131–142.

Roberts, G. O. and J. S. Rosenthal (2009). Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics 18*(2), 349–367.

Roberts, G. O. and R. L. Tweedie (1996). Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 341–363.

Rockafellar, R. and R. J.-B. Wets (1998). *Variational Analysis*. Heidelberg, Berlin, New York: Springer Verlag.

Rodríguez, A. and D. B. Dunson (2011, 03). Nonparametric bayesian models through probit stick-breaking processes. *Bayesian Anal. 6*(1), 145–177.

Rodriguez, A., D. B. Dunson, and A. E. Gelfand (2008). The nested dirichlet process. *Journal of the American Statistical Association 103*(483), 1131–1154.

Rodríguez, G., M. P. Samper, P. Ventura, L. A. Moreno, J. L. Olivares, and J. M. Pérez-González (2004, August). Gender differences in newborn subcutaneous fat distribution. *European journal of pediatrics 163*(8), 457—461.

Rodríguez, O., E. Diday, and S. Winsberg (2000). Generalization of the Principal Components Analysis to Histogram Data. pp. 12–16.

Ross, G. J. and D. Markwick (2020). *dirichletprocess: An R Package for Fitting Complex Bayesian Nonparametric Models.*

Rousseau, J. and K. Mengersen (2011). Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(5), 689–710.

Schiavon, L., A. Canale, and D. B. Dunson (2022). Generalized infinite factorization models. *Biometrika 109*(3), 817–835.

Sekitani, K. and Y. Yamamoto (1993). A recursive algorithm for finding the minimum norm point in a polytope and a pair of closest points in two polytopes. *Mathematical Programming 61*, 233–249.

Sethuraman, J. (1994). A constructive definition of dirichlet priors. *Statistica sinica*, 639–650.

Silverman, B. W. et al. (1996). Smoothed functional principal components analysis by choice of norm. *The Annals of Statistics 24*(1), 1–24.

Simon, L., P. Borrego, D. Darmaun, A. Legrand, J. Rozé, and A. Chauty-Frondas (2013). Effect of sex and gestational age on neonatal body composition. *Br J Nutr. 109*(6), 1105–8.

Soh, S.-E., M. T. Tint, P. D. Gluckman, K. M. Godfrey, A. Rifkin-Graboi, Y. H. Chan, W. Stünkel, J. D. Holbrook, K. Kwek, Y.-S. Chong, et al. (2014). Cohort profile: Growing up in singapore towards healthy outcomes (gusto) birth cohort study. *International journal of epidemiology 43*(5), 1401–1409.

Soriano, J. and L. Ma (2017). Probabilistic multi-resolution scanning for two-sample differences. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 79*(2), 547–572.

Sra, S. and I. Dhillon (2005). Generalized nonnegative matrix approximations with Bregman divergences. *Adv. Neural Inf. Process. Syst. 18.*

Srivastava, S., V. Cevher, Q. Dinh, and D. Dunson (2015a). Wasp: Scalable Bayes via barycenters of subset posteriors. In *Artificial Intelligence and Statistics*, pp. 912–920.

Srivastava, S., V. Cevher, Q. Dinh, and D. Dunson (2015b, 09–12 May). WASP: Scalable Bayes via barycenters of subset posteriors. In G. Lebanon and S. V. N. Vishwanathan (Eds.), *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, Volume 38 of *Proceedings of Machine Learning Research*, San Diego, California, USA, pp. 912–920. PMLR.

Stan Development Team (2018). Stan modeling language users guide and reference manual.

Stan Development Team (2019). *Stan Modeling Language Users Guide and Reference Manual, Version 2.26.*

Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods. *Annals of statistics*, 40–74.

Sun, S., J. Zhao, and J. Zhu (2015). A review of nyström methods for large-scale machine learning. *Information Fusion 26*, 36–48.

Symonds, M., M. Mendez, H. Meltzer, B. Koletzko, K. Godfrey, S. Forsyth, and E. van der Beek (2013). Early Life Nutritional Programming of Obesity: Mother-Child Cohort Studies. *Ann Nutr Metab 62*(2), 137–145.

Taddy, M. A., A. Kottas, et al. (2012). Mixture modeling for marked poisson processes. *Bayesian Analysis 7*(2), 335–362.

Taylor, S. J. and B. Letham (2018). Forecasting at scale. *The American Statistician 72*(1), 37–45.

Teh, Y., H. Daume III, and D. M. Roy (2007). Bayesian agglomerative clustering with coalescents. *Advances in neural information processing systems 20.*

Teh, Y. W., M. I. Jordan, M. J. Beal, and D. M. Blei (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association 101*(476), 1566–1581.

Tint, M. T., M. V. Fortier, K. M. Godfrey, B. Shuter, J. Kapur, V. S. Rajadurai, P. Agarwal, A. Chinnadurai, K. Niduvaje, Y.-H. Chan, et al. (2016). Abdominal adipose tissue compartments vary with ethnicity in asian neonates: Growing up in singapore toward healthy outcomes birth cohort study. *The American journal of clinical nutrition 103*(5), 1311–1317.

Tokdar, S. T. and R. Martin (2019). Bayesian test of normality versus a dirichlet process mixture alternative. *Sankhya B*, 1–31.

Van Der Zee, R. (2016). The 'airbnb effect': Is it real, and what is it doing to a city like amsterdam. *The Guardian 6 October 2016*.

Velázquez, E., I. Martínez, S. Getzin, K. A. Moloney, and T. Wiegand (2016). An evaluation of the state of spatial point pattern analysis in ecology. *Ecography 39*(11), 1042–1055.

Verde, R., A. Irpino, and A. Balzanella (2015, 01). Dimension reduction techniques for distributional symbolic data. *IEEE transactions on cybernetics 46*.

Villani, C. (2003). *Topics in Optimal Transportation.*

Villani, C. (2008). *Optimal Transport: old and new*, Volume 338. Springer Science & Business Media.

Wachsmuth, D. and A. Weisler (2018). Airbnb and the rent gap: Gentrification through the sharing economy. *Environment and Planning A: Economy and Space 50*(6), 1147–1170.

Wade, S., Z. Ghahramani, et al. (2018). Bayesian cluster analysis: Point estimation and credible balls (with discussion). *Bayesian Analysis 13*(2), 559–626.

Waechter, A. and L. Biegler (2006). On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming 106*, 25–56.

Walker, S., P. Damien, and P. Lenk (2004). On priors with a kullback–leibler property. *Journal of the American Statistical Association 99*(466), 404–408.

Walker, S. G. (2007). Sampling the dirichlet mixture model with slices. *Communications in Statistics—Simulation and Computation® 36*(1), 45–54.

Watanabe, S. (2013). A widely applicable Bayesian information criterion. *J. Mach. Learn. Res. 14*(Mar), 867–897.

Webster, R. and M. A. Oliver (2007). *Geostatistics for environmental scientists.* John Wiley & Sons.

Whincup, P. H., J. A. Gilg, C. G. Owen, K. Odoki, K. G. M. M. Alberti, and D. G. Cook (2005). British south asians aged 13–16 years have higher fasting glucose and insulin levels than europeans. *Diabetic Medicine 22*(9), 1275–1277.

WHO (2022). World health organization - obesity and overweigth. Accessed: 11-01-2022.

Wu, Y. and S. Ghosal (2008). Kullback leibler property of kernel mixture priors in bayesian density estimation. *Electronic Journal of Statistics 2*, 298–331.

Xie, F. and Y. Xu (2019). Bayesian repulsive gaussian mixture model. *Journal of the American Statistical Association*, 187–203.

Xu, Y., P. Müller, and D. Telesca (2016). Bayesian inference for latent biologic structure with determinantal point processes (dpp). *Biometrics 72*(3), 955–964.

Yajnik, C. S., C. H. D. Fall, K. J. Coyaji, S. S. Hirve, S. Rao, D. J. P. Barker, C. Joglekar, and S. Kellingray (2003). Neonatal anthropometry: the thin–fat Indian baby. The Pune Maternal Nutrition Study. *International Journal of Obesity 27*(2), 173–180.

Yajnik, C. S., H. G. Lubree, S. S. Rege, S. S. Naik, J. A. Deshpande, S. S. Deshpande, C. V. Joglekar, and J. S. Yudkin (2002, 12). Adiposity and Hyperinsulinemia in Indians Are Present at Birth. *The Journal of Clinical Endocrinology & Metabolism 87*(12), 5575–5580.

Zabell, S. L. (2005). *The Continuum of Inductive Methods Revisited*, pp. 243–274. Cambridge Studies in Probability, Induction and Decision Theory. Cambridge University Press.

Zemel, Y. and V. M. Panaretos (2019). Fréchet means and procrustes analysis in wasserstein space. *Bernoulli 25*(2), 932–976.

Zhang, C., P. Kokoszka, and A. Petersen (2020). Wasserstein autoregressive models for density time series. *arXiv preprint arXiv:2006.12640*.

Zhang, T., P. K. Whelton, B. Xi, M. Krousel-Wood, L. Bazzano, J. He, W. Chen, and S. Li (2019). Rate of change in body mass index at different ages during childhood and adult obesity risk. *Pediatric obesity 14*(7), e12513.

Zhou, Z., D. S. Matteson, D. B. Woodard, S. G. Henderson, and A. C. Micheas (2015). A spatio-temporal point process model for ambulance demand. *Journal of the American Statistical Association 110*(509), 6–15.