SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Department of
Economics, Management and Statistics

PhD program: **Economics, Statistics and Data Science** Cycle: **XXXVII**
Curriculum: **Statistics**

Bayesian Methods for Change Point Analysis

Surname: **DANESE**
Name: **LUCA**
Registration number: **800695**

Supervisor: Prof. **ANDREA ONGARO**
Co-Supervisor: Dott. **RICCARDO CORRADIN**

Coordinator: Prof. **MATTEO MANERA**

Academic Year: 2024/2025

Abstract

Detecting structural changes is a critical issue in many fields. Identifying when and whether a process undergoes an abrupt change can advance knowledge, support better decision-making, and improve productivity in domains such as biology, ecology, finance, and beyond. Statistical analysis addresses this problem by providing a variety of methods to detect change points in time-dependent data. Approaches based on a Bayesian framework are particularly powerful in this context, as they naturally quantify uncertainty, allow flexibility in the number and location of change points, and incorporate prior knowledge. In particular, methods based on random partitions are especially effective, as they build on the literature on Bayesian clustering to address this specific problem. This thesis advances the Bayesian methodology for change point analysis through four complementary projects. The first project develops a method for multivariate change point detection with missing observations, extending one of the leading state-of-the-art approaches by introducing a multivariate kernel and an imputation strategy within the MCMC algorithm. Applications include COVID-19 incidence in Italy, Milan temperature data, and inflation rates in the European Union. The second project introduces a model-based clustering approach to group time-dependent data with common change points. The modeling strategy is based on a mixture model, in which two observations belong to the same cluster if they share the latent partition induced by their change points. Posterior inference is performed using a split–merge algorithm. Empirical applications are provided for the spread of COVID-19 in the European Union and for exchange rate data. The third project presents a software package implementing the methods introduced in the first and second projects. The package provides efficient functions, a user-friendly interface, and tools for result visualization. The fourth project introduces a method for clustering time series with asynchronous change points. The approach is based on a repulsive finite mixture model, which encourages heterogeneous clusters to group time series that exhibit similar but not identical change points. An application to the top 50 companies in the Standard & Poor's 500 index is presented.

Acknowledgements

Before thanking those who supported me during these years, I would like to begin by expressing my gratitude to the referees for their valuable comments on this thesis.

Regarding these years, I would like to start by expressing my profound gratitude to my supervisors, Andrea Ongaro and Riccardo Corradin, for guiding me throughout this journey. Thank you for giving me the opportunity to work with you and to learn from you. I am grateful for the time you dedicated to me and the guidance you provided during these years.

I would also like to thank my colleagues Alessandro, Chiara, Luca, and Lorenzo. We shared many exams, conferences, and beers; your friendship made these years much more special. I will truly miss the time we spent together!

My thanks also go to my friends from Bicocca and elsewhere: Alice, Andrea, Anna, Ciccio, Federico, Gila, Giorgia, Jiefeng, Laura, Laura, Luca Aie, Roberto, Teo, Tommaso, and Valentina.

I would also like to mention my friends in Nottingham: Fabrizio, Kush, Olga, Wasiur, Yajie, and Yordan. I am deeply grateful to you for welcoming me during my research stay and for being such good friends throughout those six months.

I am deeply grateful to Mom, Dad, Pietro, and Maria, as well as to my family and friends, for their constant support and encouragement.

Finally, I would like to thank my wife, Miriam, for always encouraging me to pursue what makes me happy and for always being at my side. And thank you Martino - the most incredible change point for our lives so far!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The analysis of time dependent data is a key topic in statistics. Nowadays data with a temporal dependence are observed in many fields, such as finance, biology, engineering and social sciences. Many statistical methods have been developed to perform different kinds of analysis on these data. Some of them aims at describing the evolution of the sequence of observations with different dependence assumptions, other provide methods for forecasting the future behaviours of a time series. See the book by Box et al. (2016) for a comprehensive introduction to statistical methods for time series analysis. Among all the possible analysis, *change point detection* aims at finding abrupt changes that occur in the behaviour of the time series. Suppose we have a time series $\boldsymbol{y}$ composed by a set of T ordered realisations $(y_1, \dots, y_T)$ where the generic y_i, for $i = 1, \dots, T$, is distributed according to a common parametric distribution f, indexed by a time dependent parameter θ_t,

$$y_i \sim f(\cdot \mid \theta_t).$$

If $\theta_t \neq \theta_{t+1}$ then a change point occurs at time $i + 1$, corresponding to realisation y_{i+1}. In literature, we can find both frequentist and Bayesian approaches to change point detection. Despite our work completely follows a Bayesian paradigm, to give a general overview on the different methods we provide in the following two sections a literature review on both approaches.

1.1 Frequentist change point detection

Seminal contributions on frequentist change point detection are by Page (1954, 1957), who employed an approach using hypothesis tests to identify a single structural change in a time-dependent parameter. Later, such approaches have been extended in different directions by considering different methods to detect a change point. These include parametric and nonparametric approaches and cases where the position of the change point is assumed to be known or unknown (see, e.g., Brodsky and Darkhovsky, 1993). However, trying to detect a single change point might be too restrictive since in many situations more than one change might occur. Different works then extended these methods to the problem of multiple change point detection. Niu et al. (2016) provide a general review of methods for detecting multiple change points in the context of a Gaussian model. Inclan and Tiao (1994) proposed an iterative method to detect changes in the variance of a time series, where a change point is detected whenever an abrupt

change occurs in the cumulative sum of squares of the time series realisations. Similarly, Chen and Gupta (1997) proposed a method to infer the structural changes by selecting the configuration of change points that minimizes the Schwarz information criterion. See Chen and Gupta (2012) for a review on frequentist models for change point detection and possible extensions. Change points may also occur in different scenarios. For example, Young (2013) studied a mixture of regression models to estimate the distribution of the response variable, conditionally on the predictors, in the presence of possible change points in the regression effects.

1.2 Bayesian change point detection

Many contributions to change point detection can be found also in the Bayesian literature. Since literature also in this case is very broad, we only mention some of the possible approaches. The methods presented in this thesis are based on product partition models and random partitions, in the next subsection we focus in particular on literature with this kind of approach.

Early studies on Bayesian change point analysis are by Chernoff and Zacks (1964) and Kander and Zacks (1966). Their methods are based on hypothesis testings, where under the null hypothesis all the realisations of the time series are distributed according to the same distribution, while under the alternative hypothesis this distribution changes in a certain time. In this context, a few years later, Smith (1975) assumed exchangeability among observations within the same block—defined as the set of observations between two change points—and independence among observations in different blocks. This assumption can be found also on approaches based on product partition models (see Section 1.2.1). Later, Yao (1984) proposed a Bayesian model for estimating a step function with additive Gaussian noise in the presence of multiple change points.

Several Bayesian approaches were developed to perform change point detection in more complex settings. Among these, Chib (1998) introduces a method in which a latent discrete variable, driven by a Markov process, indicates to which regime the parameter belongs. Following this approach Peluso et al. (2019) propose a semiparametric model where the parameter can be multivariate, and subject to change points at different times. Nonparametric Bayesian models have also been proposed, Muliere and Scarsini (1985) for example relaxed the parametric assumption on f by assuming a Dirichlet process prior for the distributions before and after a single change point. Successively, Mira and Petrone (1997) assumed dependencies among the distributions before and after the change point using as prior distribution a mixture of products of Dirichlet processes.

The same change points might also occur in different time series, but not simultaneously. For example, if a group of sensors is detecting earthquakes, we expect the same signal to be detected by all the sensors with a short time lag. To address this problem, Hallgren et al. (2024) and McKee and Kalli (2025) propose both a method for detecting asynchronous change points in which the dependencies among the time series are modelled by a latent graph.

1.2.1 Random partitions and Product Partition Models in the context of change point detection

Recently, the problem of change point detection has been formulated also in terms of Product Partition Model (PPM). The methods presented in this thesis are all based on this approach.

Product Partition Models are a family of models introduced by Hartigan (1990) in the context of clustering analysis. Given a set of n observations $(y_1, \dots, y_n)$, a partition of these in k clusters can be represented by an object $\rho = \{A_1, \dots, A_k\}$, where the general A_i is the set of all observations that belong to the $i-$th block. The idea behind PPMs is to consider the partition as a random variable distributed according to the following probability:

$$p(\rho = \{A_1, \dots, A_k\}) = K \prod_{j=1}^{k} c(A_j), \tag{1.1}$$

where $c(A_j) \geq 0$ is a function called cohesion function that quantifies how cohesive are the observations in cluster j, and K denotes the normalisations constant given by:

$$K = \left(\sum_{k=1}^{n} \sum_{\rho \in \mathcal{P}_{n,k}} \prod_{j=1}^{k} c(A_j) \right)^{-1},$$

where $\mathcal{P}_{n,k}$ is the space of all possible partitions of n elements in k groups. The partition ρ is then renamed *random partition* to emphasize its nature of random variable.

Indeed, the problem of change point detection can be considered as a problem of clustering where the observations before and after a change points belong to different clusters. The use of PPMs for change point detection was first proposed by Barry and Hartigan (1992), who introduced a cohesion function that encourages large blocks and assigns probability zero to blocks consisting of observations that are not contiguous in time. Later, inspired by Yao (1984), Barry and Hartigan (1993) propose the following cohesion function, where the probability q of change point occurrence is the same at every time instant,

$$c(A_j) = \begin{cases} (1-q)\, q^{\,m_j - 1}, & \text{if } j < n, \\ (1-q)^{\,m_j - 1}, & \text{if } j = n, \end{cases}$$

where m_j is the number of observation in block A_j and $0 \leq q \leq 1$. While these first works are based on a single parameter-usually the mean on a Gaussian kernel-later Loschi et al. (2003) and Loschi and Cruz (2002) propose to extend the method by Barry and Hartigan (1993) both to possible changes in the mean and in the variance. Pedroso et al. (2023) later consider a multi-partition model where each parameter has a specific random partition, by doing so it is possible not only to detect change points on multiple parameters but also to infer on which parameters the change occurs. In all of these models a parametric kernel is assumed for the data, García and Gutiérrez-Peña (2019) instead propose to use a random measure for the distribution of the observations. Recently Quinlan et al. (2024) have addressed the problem of inducing correlation among change points in different time series. They propose a PPM-based approach

where the cohesion function by Yao (1984) is modified in order to induce correlation among the random partitions of the time series. Dependency can also be introduced directly among random partitions. Page et al. (2022) propose a method where the dependency through a temporal sequence of partitions is directly modeled by using an auxiliary variable that denote if an observation might change cluster from one time to another. Later, this approach was extended by Paganin et al. (2025) in order to include prior knowledge coming from an initial partition. Although these models have not been applied to change point problems, such an application could be of great interest. Change points can also be detected among partitions, Giampino et al. (2024) propose a method to detect changes on a sequence of random partitions using a partition-based state equation that model the temporal dynamic of the clustering structures. Partitions in this context can have both a spatial and temporal dimension, like in Cremaschi et al. (2023), where a semiparametric Bayesian model is proposed to cluster the areas of the city of Milan (Italy) depending on the mobile phone usage throughout different times of the day. Since these clusters may change over time, their model also accounts for change point detection on these random partitions.

The PPM that describes the random partition can also be chosen from the class of Bayesian nonparametric (BNP) prior distributions called *Gibbs-type priors*. In this class the probability of the partition induced by a species sampling model in a set of exchangeable observations is called the *Exchangeable Partition Probability Function* (EPPF). If an EPPF arises from a Gibbs-type prior then it can be written as a PPM, as shown by Quintana and Iglesias (2003) in the case of the Dirichlet process. As byproduct, the probability of a partition ρ inherit the property that it depends only on the cardinalities of the blocks

$$p(\rho = \{A_1, \dots, A_k\}) = p(m_1, \dots, m_k),$$

where m_i is the cardinality of group A_i. In change point detection problems, not all possible partitions of a set of temporal observations are admissible. The admissible partitions are only those in which each block contains consecutive observations. For example, the partition $\rho = \{\{y_1, y_2, y_4\}, \{y_3\}\}$ is not admissible since y_4 occurs after y_3. Therefore, when dealing with random partitions in the context of change point detection, we should instead consider *random orders*, where orders are a subset of partitions that respect the temporal ordering. Fuentes–García et al. (2010) propose adapting the EPPF from a Dirichlet process to random orders by assigning probability zero to those partitions that are not admissible. Given a sequence of T time realisations $(y_{t_1}, \dots, y_{t_T})$ grouped into k ordered blocks $(A_1, \dots, A_k)$, they define a restricted EPPF $\widetilde{p}$, obtained by modifying the EPPF p of a Dirichlet process in the following way:

$$\widetilde{p}(m_1, \dots, m_k) \propto p(m_1, \dots, m_k)\, \mathbb{1}(t_1 < t_2 < \cdots < t_T).$$

Later, following the work by Pitman (2006), Martínez and Mena (2014) considered a transformation of the EPPF arising from a Pitman–Yor process (Pitman, 1995), a more general prior distribution that includes the Dirichlet process as a special case. This transformation preserves the key properties of the process, such as the symmetry of the corresponding probability distribution and the distribution of the number of distinct blocks, but restricts the support of the

function to the space of admissible partitions. Further details about this approach are provided in Chapter 2.

1.3 Contributions and structure of the thesis

This thesis is structured in four strongly interconnected projects that propose new developments in the context of Bayesian change point analysis.

The first project (Chapter 2) is a method for change point detection on multivariate data with missing observations. We extend the work by Martínez and Mena (2014) by incorporating a multivariate kernel for the time series and a procedure for missing data imputation inside the Monte Carlo Markov Chain (MCMC) algorithm. We illustrate our method through three applications: the proportion of COVID-19 new daily cases in Italy, temperature data in the city of Milan, and inflation rates in European Union.

The second project (Chapter 3) proposes a novel idea: clustering samples of time-dependent observations where we assume as unique commonality that two observations belong to the same group if structural changes in their behaviours happen at the same time. These changes are modeled according to the same theory presented in Chapter 2. The posterior sampling strategy is an informed split and merge algorithm designed to accommodate the discrete nature of the data involved. We show an application to the spreading processes of the COVID-19 virus in the countries of European Union and to financial data.

The third project (Chapter 4) is an R package, built in C++, called **BayesChange**. This package provides methods to perform change point detection and clustering of observations sharing common change points according to the models introduced in Chapter 2 and 3. The core functions are implemented in C++ to ensure computational efficiency, while an R user interface simplifies the package usage. The **BayesChange** package includes two R wrappers that integrate the C++ backend functions, along with S3 methods for summarizing the results both numerically and graphically. We briefly present the theory beyond each method and the algorithms for posterior simulation and provide details about the arguments, the inputs and the outputs of each function. Finally we illustrate through synthetic examples the package's usage.

The fourth project (Chapter 5), showing preliminary but promising results, proposes an alternative to the clustering model introduced in Chapter 4. In this approach, time series within the same cluster share the same change points, although their exact locations may differ slightly. To achieve this, we propose a finite mixture model with a fixed number of components and a repulsive mechanism. We conduct a simulation study to investigate the properties of the posterior distribution and apply the model to the stock prices of the top 50 companies by market capitalization in the Standard & Poor's 500 index.

The first and second projects resulted in the following published papers: Corradin et al. (2022) and Corradin et al. (2026). The paper based on the third project has been recently submitted.

Chapter 2

Bayesian Nonparametric Change Point Detection for Multivariate Time Series with Missing Observations

2.1 Introduction

In this project, we propose a Bayesian nonparametric model for change point detection in multivariate data. Although most of the literature on change point analysis focuses on univariate time series, several approaches have been proposed for the multivariate setting. For example, Chen et al. (2000) provide a gentle introduction, while Li et al. (2019) propose a minimum description length procedure to detect change points in multivariate time series. See also Truong et al. (2020) for a recent review of multiple change point detection methods for multivariate time series.

Our proposal is an extension of the model introduced by Martínez and Mena (2014) to the multivariate case, working directly with a latent combinatorial structure in the data. Our main contribution is in terms of results for dealing efficiently, and possibly in the presence of missing realisations, with change point analyses for a multivariate time series. We use a multivariate Ornstein–Uhlenbeck process to describe the distribution of the data within each block of the time series. This chapter is organized as follows: Section 2.2 provides a formal introduction to the modeling framework, and a discussion of the key quantities needed to perform change point detection in a multivariate regime. Section 2.3 provides insights into the proposed model by investigating its properties via a simulation study. Section 2.4 reports the posterior summaries of the model estimates for two real scenarios: the study of the change points in the composition of the daily new cases of coronavirus disease 2019 (COVID-19) in different areas of Italy, the study of change points in the time series of 27 yearly temperature curves measured in Milan (Italy) and finally inflation rates in the European Union. Section 2.5 presents the final remarks and discussion. The proofs of the main results are provided in Appendix A, as well as the main routines to perform posterior inference.

2.2 Modelling multiple change points in a multivariate time series

We retain $\boldsymbol{y} = (y_{t_1}, \dots, y_{t_T})$ a time series indexed by t_i, where $i = 1, \dots, T$ and with the generic y_{t_i} taking values in $\mathbb{Y} \subseteq \mathbb{R}^d$. We endeavor to model the distribution of possible latent change points of the time series from a Bayesian nonparametric perspective. We assume the generic $i-$th element of the time series distributed according to a law $\mathcal{L}(y_{t_i} \mid \theta_{t_i}, \dots)$ where θ_{t_i} is a parameter indexing the distribution of y_{t_i}, and this distribution may depend on other quantities, such as previous realisations of the time series. We further assume that the generic θ_{t_i} takes values on $(\Theta, \mathcal{T})$, with Θ Polish space and $\mathcal{T}$ denoting its Borel σ-field.

Let $\mathcal{C}_T$ denote the set of possible orders of $[T] = \{1, \dots, T\}$, that is, a partition of the set $[T]$ into k distinct groups $A_1, \dots, A_k$, with $1 \leq k \leq T$, where $A_j \cap A_l = \emptyset$, $\bigcup_{j=1}^{k} A_j = [T]$, and the sets $A_1, \dots A_k$ are subject to a monotone increasing constraint of their elements, such that if $j \in A_j$ and $l \in A_k$ with $k > j$, then $l > j$. We further denote a generic random order of $[T]$ into k distinct blocks by $\rho = (A_1, \dots, A_k)$, with $\rho \in \mathcal{C}_T$. In a model-based regime, we associate each block with a specific value of the latent parameters, shared by all the observations belonging to such block. We then denote by $\theta^*_{t_1}, \dots \theta^*_{t_k}$ the specific value of the parameters for k distinct blocks, with the generic $\theta^*_{t_j}$ taking values in $(\Theta, \mathcal{T})$, with $\theta_{t_i} = \theta^*_{t_j}$ for all indices $t_i \in A_j$. A reasonable assumption to model the latent random order in the data parameters is that the distribution of the order is independent of the distribution of the block-specific parameters $\theta^*_{t_1}, \dots \theta^*_{t_k}$, that is the specific shape of the time series within each block does not depend on the process governing the behavioural changes. Within this assumption, we can write the distribution of the latent parameters as the product of two distinct terms, with

$$\mathcal{L}(\theta_{t_1}, \dots, \theta_{t_T}) = \mathcal{L}(\rho)\mathcal{L}(\theta^*_{t_1}, \dots, \theta^*_{t_k}) \tag{2.1}$$

In the spirit of the seminal studies by Barry and Hartigan (1992), we can further specify the distribution of the latent order of the data in a product partition fashion, as

$$\mathcal{L}(\rho) = K \prod_{j=1}^{k} c(A_j), \tag{2.2}$$

where $K^{-1} = \sum_{\rho \in \mathcal{C}_T} \prod_{j=1}^{k} c(A_j)$ is a normalisation constant, and $c(\cdot)$ denotes the cohesion function, that acts on the random order through blocks A_j's. Several choices of $c(\cdot)$ can be made to transform Equation (2.2) into a proper distribution, taking values on $\mathcal{C}_T$. A first attempt at such strategy was earlier introduced by Fuentes–García et al. (2010), who used a restriction of EPPFs to space $\mathcal{C}_T$ by equally penalizing all the elements in $\mathcal{C}_T$. Later, Martínez and Mena (2014) extended this modeling approach to the case where K and $c(\cdot)$ were fully specified by restricting the EPPF arising from a Pitman-Yor process (Pitman, 2006) to $\mathcal{C}_T$, but using a correction that may differ for different elements of $\mathcal{C}_T$. They introduced the restricted

EPPF of the Pitman-Yor process as

$$\mathcal{L}(\rho) = p_k^{(T)}(m_1, \dots, m_k) = \frac{T!}{k!} \frac{\prod_{j=1}^{k-1}(\delta + j\sigma)}{(\delta+1)_{T-1}} \prod_{j=1}^{k} \frac{(1-\sigma)_{m_j-1}}{m_j!} \tag{2.3}$$

where k is the number of blocks in ρ, m_j denotes the cardinality of the j-th block of $\boldsymbol{y}$, $(a)_b$ (with $a, b > 0$) denotes the Pochammer symbol, $\sigma \in [0, 1)$ is the discount parameter acting on the cardinality of each block, and $\delta \in (-\sigma, \infty)$ denotes the strength parameter. Instead of other possible ways to restrict an EPPF to $\mathcal{C}_T$, a restriction as in Equation (2.3) is preserving not only the combinatorial structure, but also other features of the starting EPPF, such as the prior expected number of distinct blocks in the order (see Lemma 1 of Martínez and Mena (2014)).

We further assume that $\theta^*_{t_1}, \dots, \theta^*_{t_k}$ given ρ are a priori independent and identically distributed from a common distribution P_0 diffuse on Θ, with

$$\mathcal{L}(\theta^*_1, \dots, \theta^*_{t_k}) = \prod_{j=1}^{k} P_0(\theta^*_{t_j}).$$

We assume the distribution of the i-th realisation in the $j-$th part of the time series depending on $\theta^*_{t_j}$. The distribution of $y_{t_1}, \dots y_{t_T}$ and $\theta_{t_1}, \dots, \theta_{t_T}$, divided into k distinct blocks described by ρ, can be written as

$$\begin{aligned} \mathcal{L}(y_{t_1}, \dots y_{t_T}, \theta_{t_1}, \dots, \theta_{t_T}) &= \mathcal{L}(y_{t_1}, \dots y_{t_T} \mid \rho, \theta_{t_1}, \dots, \theta_{t_T}) \mathcal{L}(\rho) \mathcal{L}(\theta^*_{t_1}, \dots, \theta^*_{t_k}) \\ &= p_k^{(T)}(m_1, \dots, m_k) \prod_{j=1}^{k} [P_0(\theta^*_{t_j}) \mathcal{L}(\{y_{t_i} : t_i \in A_j\} \mid \theta^*_{t_j})], \end{aligned} \tag{2.4}$$

where $p_k^{(T)}(m_1, \dots, m_k)$ is the restricted EPPF of a Pitman-Yor process to $\mathcal{C}_T$ as in Equation (2.3), $\mathcal{L}(\{y_{t_i} : i \in A_j\} \mid \theta^*_j)$ plays the role of the likelihood in the $j-$th block of data, and $\theta^*_{t_j}$ denotes the parameter indexing the distribution of the $j-$th block of the time series. Within a Markovian regime, with the generic y_{t_i} depending on the past only through $y_{t_{i-1}}$, the law of the generic $j-$th block of data becomes

$$\mathcal{L}(\{y_{t_i} : t_i \in A_j\} \mid \theta^*_{t_j}) = \mathcal{L}(y_{t_{h_j+1}} \mid \theta^*_{t_j}) \prod_{i=h_j+2}^{h_j+m_j} \mathcal{L}(y_{t_i} \mid y_{t_{i-1}}, \theta^*_{t_j}),$$

where $h_j = \sum_{l=1}^{j-1} m_l$. We aim to identify blocks $A_1, \dots, A_k$ efficiently in a multivariate scenario. We propose an extension of the model presented by Martínez and Mena (2014) to the multivariate case, accommodating multivariate kernel functions for time series data, with possibly non-equally spaced observation times.

2.2.1 Bayesian inference with multivariate Ornstein-Uhlenbeck process

We now focus on the distribution of a specific set of realisations belonging to the same block of the time series, which we assume is distributed according to a multivariate Ornstein-Uhlenbeck process. The Ornstein-Uhlenbeck process, introduced by Uhlenbeck and Ornstein (1930), is a

continuous time diffusion process taking values on $\mathbb{R}$ deeply studied in the last decades, and extended in several directions. In the univariate case, for $y_t \in \mathbb{R}^d$ where $d = 1$, the process is defined by the stochastic differential equation

$$dy_t = -\beta(y_t - \mu)dt + \tau dW_t, \tag{2.5}$$

where $\beta > 0$ and $\tau > 0$; $\mu \in \mathbb{R}$ is the long-term mean of the process, and W_t denotes a one-dimensional Wiener process. A distinctive feature of the Ornstein–Uhlenbeck process is that it is stationary, reversible, and Gaussian. It is widely used in finance, making it a suitable model for our change point detection applications. One of the main extensions of the Ornstein–Uhlenbeck process considered in the literature is the multivariate case, leading to a diffusion process taking values in $\mathbb{R}^d$. The multivariate Ornstein–Uhlenbeck process is defined by the stochastic differential equation

$$dy_t = B(y_t - \boldsymbol{\mu})dt + \Sigma d\boldsymbol{W}_t, \tag{2.6}$$

where $y_t \in \mathbb{R}^d$ with $d > 1$. B and Σ are matrices taking values on $M_{d\times d}$, with $M_{d\times d}$ denoting the space of real-valued matrices with dimensions $d \times d$, $\boldsymbol{\mu} \in \mathbb{R}^d$, with Σ as an invertible matrix, and $\boldsymbol{W}_t$ representing a $d-$dimensional Wiener process. The solution of (2.6) leads to the following conditional distribution

$$y_t \mid y_s = \widetilde{y}_s \sim \mathrm{N}(\boldsymbol{\mu} + e^{B|t-s|}(\widetilde{y}_s - \boldsymbol{\mu}), \Lambda - e^{B|t-s|}\Lambda e^{B|t-s|}), \tag{2.7}$$

where $\widetilde{y}_s \in \mathbb{Y}$, $\Lambda = \Sigma\Sigma^\top$, and by assuming a regularity condition on matrix B, that is, B is a stable matrix, the equilibrium distribution of y_t is given by

$$y_t \sim \mathrm{N}(\boldsymbol{\mu}, \Lambda).$$

We hereby consider the isotropic case where $B = bI_d$, with I_d corresponding to the identity matrix of order d. By isotropy we mean that the dependence of the time series is uniform over the possible directions, as long as the process evolves. While such an assumption might look strong for some real applications, when we deal with dimensions defined on a similar scale, it is reasonable to assume that the process has a uniform dynamic across different directions. To reduce the impact of this assumption, one can marginally standardize the multivariate time series before carrying out the analysis, such that possible dependencies across different dimensions are preserved, but the magnitudes of the dimensions are commensurable. Note that within this assumption, it is sufficient that $b < 0$ to have $\lim_{t\to\infty} e^{B|t-s|} = 0$, so that $B = bI_d$ with $b < 0$ is a stable matrix. We set $|t - s| = \Delta$, with the generic $\Delta_i = |t_i - t_{i-1}|$ denoting the $i-$th time increment out of a set of $i = 1, \ldots, T$ observations, and we introduce a suitable reparameterization of the distribution $y_t \mid y_s = \widetilde{y}_s$. By setting $\gamma = e^b$, we have that $\Psi^\Delta = e^{b\Delta} = \gamma^\Delta I_d$ and

$$Var(y_t \mid y_{t-s} = \widetilde{y}_{t-s}) = \Lambda - \Psi^\Delta \Lambda \Psi^\Delta = \Lambda(1 - \gamma^{2\Delta}),$$

obtaining the conditional distribution $y_t \mid y_{t-s}$

$$y_t \mid y_{t-s} = \widetilde{y}_{t-s} \sim \mathrm{N}(\boldsymbol{\mu} + \gamma^{\Delta}(y_{t-s} - \boldsymbol{\mu}), \Lambda(1 - \gamma^{2\Delta})), \tag{2.8}$$

where $\gamma \in (0, 1)$ in force of the stability assumption $b < 0$. We denote an Ornstein-Uhlenbeck process as in Equation (2.6), with parametrization as in Equation (2.8), as $\mathrm{OU}(\boldsymbol{\mu}, \Lambda, \gamma)$. We aim to infer an Ornstein-Uhlenbeck process from a Bayesian perspective. Examples of the use and study of the multivariate Ornstein-Uhlenbeck process within a Bayesian framework can be found in various fields, such as Blackwell (2003), Singh et al. (2018), and the references within. We set a normal-Inverse-Wishart prior (NIW) $\pi(\boldsymbol{\mu}, \Lambda)$ for the joint distribution of $\boldsymbol{\mu}$ and Λ, such that $(\boldsymbol{\mu}, \Lambda) \sim \mathrm{NIW}(\boldsymbol{m}_0, \kappa_0, \nu_0, S_0)$, specified as

$$\begin{aligned} \Lambda &\sim \mathrm{IW}(\nu_0, S_0) \\ \boldsymbol{\mu} \mid \Lambda &\sim \mathrm{N}(\boldsymbol{m}_0, \frac{\Lambda}{\kappa_0}), \end{aligned} \tag{2.9}$$

which, as a conjugate distribution, leads to a simple form of the posterior quantities. The previous are distributional assumptions arising from the Gaussian marginal distribution of single realisations of the process, which are commonly used in the case of independent Gaussian realisations. In contrast to setting an independent prior on the parameters $\boldsymbol{\mu}$ and Λ, which leads to marginal conjugacy, the specification in Equation (2.9) is jointly conjugate for the parameters $(\boldsymbol{\mu}, \Lambda)$. In the following proposition, we characterize the posterior distribution of $\boldsymbol{\mu}$ and Λ, given $y_{t_1}, \dots, y_{t_{m_j}}$ and γ, where $y_{t_1}, \dots, y_{t_{m_j}}$ denotes the set of observations belonging to the generic j-th block of the time series.

Proposition 1 *Let $y_{t_1}, \dots, y_{t_{m_j}}$ be a set of realisations from an isotropic and stable $OU(\boldsymbol{\mu}, \Lambda, \gamma)$ process, with time increments $\Delta_1, \dots, \Delta_{m_j}$. We assume a priori $(\boldsymbol{\mu}, \Lambda)$ distributed according to a NIW distribution, $(\boldsymbol{\mu}, \Lambda) \sim NIW(\boldsymbol{m}_0, \kappa_0, \nu_0, S_0)$. Then the posterior distribution, conditional on γ, is a $NIW(\boldsymbol{m}_n, \kappa_n, \nu_n, S_n)$, with*

$$\begin{aligned} \kappa_n &= \kappa_0 + \sum_{i=2}^{m_j} \frac{(1 - \gamma^{\Delta_i})^2}{(1 - \gamma^{2\Delta_i})} + 1 \\ \boldsymbol{m}_n &= \frac{1}{\kappa_n} \left[\kappa_0 \boldsymbol{m}_0 + y_{t_1} + \sum_{i=2}^{m_j} \frac{(1 - \gamma^{\Delta_i})}{(1 - \gamma^{2\Delta_i})} (y_{t_i} - \gamma^{\Delta_i} y_{t_{i-1}}) \right] \\ \nu_n &= \nu_0 + m_j \\ S_n &= S_0 + y_{t_1} y_{t_1}^{\top} + \sum_{i=2}^{m_j} \frac{(y_{t_i} - \gamma^{\Delta_i} y_{t_{i-1}})(y_{t_i} - \gamma^{\Delta_i} y_{t_{i-1}})^{\top}}{(1 - \gamma^{2\Delta_i})} + \kappa_0 \boldsymbol{m}_0 \boldsymbol{m}_0^{\top} - \kappa_n \boldsymbol{m}_n \boldsymbol{m}_n^{\top} \end{aligned}$$

From Proposition 1 we have an explicit form of the parameters indexing the posterior distribution of $(\boldsymbol{\mu}, \Lambda)$, conditional on γ. Such a distribution is still a NIW distribution, but it incorporates the empirical information tempered by the dependence parameter γ. Because we aim to perform inference on the latent change points in a time series, for most of the inferential purposes of our analysis, we are not interested in collecting realisations from the posterior distribution of the parameters $\boldsymbol{\mu}$ and Λ. Indeed, conditional on a specific allocation of the data in different blocks,

we are more interested in a measure of the marginal likelihood of the data rather than the values of the group-specific parameter. To this end, we can obtain a tractable form for the integrated marginal likelihood of $y_{t_1}, \dots, y_{t_{m_j}} \mid \gamma$ as follows.

Proposition 2 *Let $y_{t_1}, \dots, y_{t_{m_j}}$ be a set of realisations from an isotropic and stable OU$(\boldsymbol{\mu}, \Lambda, \gamma)$ process, with time increments $\Delta_1, \dots, \Delta_{t_{m_j}}$. Let $\pi(\boldsymbol{\mu}, \Lambda)$ denotes the prior on $(\boldsymbol{\mu}, \Lambda)$, a NIW$(\boldsymbol{m}_0, \kappa_0, \nu_0, S_0)$ distribution. Then the integrated marginal likelihood of $y_{t_1}, \dots, y_{t_{m_j}}$ given γ is equal to*

$$\mathcal{L}(y_{t_1}, \dots, y_{t_{m_j}} \mid \gamma) = \frac{\kappa_n^{-\frac{d}{2}}}{\kappa_0^{-\frac{d}{2}}} \frac{\pi^{-\frac{m_j d}{2}}}{\prod_{i=2}^{m_j} (1-\gamma^{2\Delta_i})^{\frac{1}{2}}} \frac{|S_0|^{\frac{\nu_0}{2}}}{\Gamma_d(\frac{\nu_0}{2})} \frac{\Gamma_d(\frac{\nu_n}{2})}{|S_n|^{\frac{\nu_n}{2}}}, \tag{2.10}$$

with κ_n, ν_n and S_n specified according to Proposition 1.

By exploiting the integrated marginal likelihood of Equation (2.10), we can obtain efficient sampling strategies when we are not interested in specific inferences for the parameters characterizing the process. Detailed proofs of Proposition 1 and 2 are provided in the Appendix. Real cases of this study may show missing observations. One possible strategy to deal with missing data is to perform data imputation. We can easily derive the distribution of the generic y_{t_j} given $y_{(t_j)} = \{y_{t_1}, \dots, y_{t_{m_j}}\} \setminus y_{t_j}$.

Proposition 3 *Let $y_{t_1}, \dots, y_{t_{m_j}}$ be a set of realisations from an isotropic and stable OU$(\boldsymbol{\mu}, \Lambda, \gamma)$ process, with time increments $\Delta_1, \dots, \Delta_{m_j}$. Then the condition distribution of $y_{t_j} \mid y_{(t_j)}, \gamma$ is a multivariate Gaussian distribution $y_{t_j} \mid y_{(t_j)}, \gamma \sim N(\boldsymbol{\lambda}, \phi)$, with:*

- *if $j = 1$*

$$\boldsymbol{\lambda} = \gamma^{\Delta_2} y_{t_2} + (1-\gamma^{\Delta_2})\boldsymbol{\mu}, \quad \phi = \Lambda(1-\gamma^{2\Delta_2});$$

- *if $1 < j < T$*

$$\boldsymbol{\lambda} = \frac{\gamma^{\Delta_j}(1-\gamma^{2\Delta_{j+1}}) y_{t_{j-1}} + \gamma^{\Delta_{j+1}}(1-\gamma^{2\Delta_j}) y_{t_{j+1}} + (1-\gamma^{\Delta_j})(1-\gamma^{2\Delta_{j+1}})\boldsymbol{\mu} - \gamma^{\Delta_{j+1}}(1-\gamma^{\Delta_{j+1}})(1-\gamma^{2\Delta_j})\boldsymbol{\mu}}{1-\gamma^{2(\Delta_j+\Delta_{j+1})}}$$

$$\phi = \Lambda\left(\frac{(1-\gamma^{2\Delta_j})(1-\gamma^{2\Delta_{j+1}})}{1-\gamma^{2(\Delta_j+\Delta_{j+1})}}\right);$$

- *if $j = m_j$*

$$\boldsymbol{\lambda} = \gamma^{\Delta_{m_j}} y_{t_{T-1}} + (1-\gamma^{\Delta_{m_j}})\boldsymbol{\mu}, \quad \phi = \Lambda(1-\gamma^{2\Delta_{m_j}});$$

The proof of Proposition 3 is provided in Appendix A. Notably, Proposition 3 is not restricted to performing data imputation, but it is also an explicit expression for intra-time prediction. We can further quantify the dependence across different dimensions over distinct observations times in the isotropic case, as stated in the following proposition.

Proposition 4 *Let $y_{t_{i-1}}$ and y_{t_i} be two distinct realisations of an isotropic and stable OU$(\boldsymbol{\mu}, \Lambda, \gamma)$ process, with time increments Δ_i. Then we have*

$$Cov(y_{t_i}, y_{t_{i-1}}) = \gamma^{\Delta_i} Var(y_{t_{i-1}}).$$

A sketch of the proof of Proposition 4 is presented in Appendix A. From Proposition 4 we can appreciate how even the isotropic case, a dependence across different dimensions is preserved over different observations times, proportional to the intrinsic covariance of the data, but tempered by the dependence parameter γ. Proposition 1,2 and 3 describe all the key quantities that characterize the parametric part of the model introduced in the first part of Section 2.2. Although we derive the previous quantities for the scenario with different time increments, it is common to observe equally spaced time series, which corresponds to the case with $\Delta_i = c$ for all $i = 1, \ldots, m_j$. Without loss of generality, we can assume that $c = 1$. Within the equally spaced time scenario, we can derive simpler forms of the expressions in Propositions 1,2 and 3. The posterior distribution of $\boldsymbol{\mu}$ and Λ, assuming a NIW$(\boldsymbol{m}_0, \kappa_0, \nu_0, S_0)$ prior and an isotropic and stable Ornstein-Uhlenbeck process likelihood, with equally spaced time increments and conditionally on γ, is a NIW$(\boldsymbol{m}_n, \kappa_n, \nu_n, S_n)$, with

$$
\begin{aligned}
\kappa_n &= \kappa_0 + (m_j - 1)\frac{(1-\gamma)^2}{(1-\gamma^2)} + 1, \\
\boldsymbol{m}_n &= \frac{1}{\kappa_n}\left[\kappa_0 \boldsymbol{m}_0 + y_{t_1} + \frac{(1-\gamma)}{(1-\gamma^2)}\sum_{i=2}^{m_j}(y_{t_i} - \gamma\, y_{t_{i-1}})\right], \\
\nu_n &= \nu_0 + m_j, \\
S_n &= S_0 + y_{t_1} y_{t_1}^\top + \frac{1}{(1-\gamma^2)}\sum_{i=2}^{m_j}(y_{t_i} - \gamma\, y_{t_{i-1}})(y_{t_i} - \gamma\, y_{t_{i-1}})^\top + \kappa_0 \boldsymbol{m}_0 \boldsymbol{m}_0^\top - \kappa_n \boldsymbol{m}_n \boldsymbol{m}_n^\top.
\end{aligned}
\tag{2.11}
$$

Similarly, we can express the integrated marginal likelihood for the equally spaced time case to obtain

$$
\mathcal{L}(y_{t_1}, \ldots, y_{m_j} \mid \gamma) = \frac{\kappa_n^{-\frac{d}{2}}}{\kappa_0^{-\frac{d}{2}}} \frac{\pi^{-\frac{m_j d}{2}}}{(1-\gamma^2)^{\frac{m_j - 1}{2}}} \frac{|S_0|^{\frac{\nu_0}{2}}}{\Gamma_d(\frac{\nu_0}{2})} \frac{\Gamma_d(\frac{\nu_n}{2})}{|S_n|^{\frac{\nu_n}{2}}},
\tag{2.12}
$$

where κ_n, ν_n and S_n correspond to the previous expression for the equally spaced time increment scenario. Furthermore, the distribution of a single observation, conditionally the rest of the sample, is again a multivariate Gaussian distribution, $y_j \mid y_{(j)}, \gamma \sim \mathrm{N}(\boldsymbol{\lambda}, \phi)$, with parameters depending on the observation's index

- if $j = 1$

$$\boldsymbol{\lambda} = \gamma\, y_{t_2} + (1-\gamma)\boldsymbol{\mu}, \qquad \phi = \Lambda(1-\gamma^2);$$

- if $1 < j < T$

$$\boldsymbol{\lambda} = \frac{\gamma\,(y_{t_{j+1}} + y_{t_{j-1}})}{(1+\gamma^2)} + \frac{(1+\gamma)^2\boldsymbol{\mu}}{(1+\gamma^2)}, \qquad \phi = \Lambda\left(\frac{1-\gamma^2}{1+\gamma^2}\right);$$

- if $j = m_j$

$$\boldsymbol{\lambda} = \gamma\, y_{m_j - 1} + (1-\gamma)\boldsymbol{\mu}, \qquad \phi = \Lambda(1-\gamma^2).$$

We can then incorporate the previous quantity within a time series model-based clustering environment, to perform multiple change point detection for a multivariate time series.

2.2.2 Prior to posterior inference

Hereafter, we consider a scenario with equally spaced time data. We assume a model that incorporates an unknown number of change points by exploiting a prior over the possible random ordering, as in Equation (2.3). We further assume that within each block of the time series, sharing the same regime, an isotropic and stable multivariate Ornstein-Uhlenbeck process describes the distributions of the observations. According to the model formulation described in Equation (2.4), by setting $\theta_i = (\boldsymbol{\mu}_i, \Lambda_i)$, we obtain a model in its hierarchical form as follows:

$$\begin{aligned} \{y_{t_i} : t_i \in A_j\} \mid \boldsymbol{\mu}_j^*, \Lambda_j^*, \gamma &\sim \mathrm{OU}(\boldsymbol{\mu}_j^*, \Lambda_j^*, \gamma) \\ \boldsymbol{\mu}_j^*, \Lambda_j^* &\overset{\text{iid}}{\sim} P_0 \\ \rho_T = (A_1, \ldots, A_k) &\sim \mathcal{L}(\rho; \delta, \sigma) \\ (\delta, \sigma, \gamma) &\sim \pi(\delta, \sigma, \gamma), \end{aligned} \tag{2.13}$$

where P_0 is a diffuse measure with support $\mathbb{R}^d \times M_{d\times d}$, and $\mathcal{L}(\rho)$ is equal to the EPPF of a Pitman-Yor process restricted to $\mathcal{C}_T$, as shown in Equation (2.3). We assume that γ is shared across different parts of the time series, and we can possibly relax the model specification by letting the main parameters of the model (δ, σ, γ) random and distributed according to a prior distribution $\pi(\delta, \sigma, \gamma)$. It is also reasonable to assume $\gamma \sim \mathrm{Beta}(a_\gamma, b_\gamma)$, $\sigma \sim \mathrm{Beta}(a_\sigma, b_\sigma)$ and $\delta \mid \sigma \sim \mathrm{sGamma}(-\sigma, a_\delta, b_\delta)$, where $\mathrm{sGamma}(-\sigma, \cdot, \cdot)$ denotes the shifted Gamma distribution by a value $-\sigma$. See Martínez and Mena (2014) for details on the posterior sampling of σ and δ within this prior setup. The posterior inference of a model as in Equation (2.13) can be conducted by collecting a sample via Markov Chain Monte Carlo (MCMC) strategies. First, we note that, as far as we are not interested in any specific value for the parameters characterizing different parts of the time series, we can use an integrated marginal likelihood approach of Equation (2.12), in the spirit of Neal (2000), Favaro and Teh (2013) and Martínez and Mena (2014). Furthermore, owing to the natural ordering of the data, we resort to a split and merge strategy. The sampling scheme we use to collect a sample from the posterior distribution is fully inspired by the algorithm described in Martínez and Mena (2014), but extended with a multivariate Ornstein-Uhlenbeck process and possibly missing observations. For the sake of completeness, we report the pseudocode in Appendix A. Once we produce a sample of random orders from the posterior distribution, we can obtain point estimates by exploiting a decision theoretic approach based on a specific loss function, solving an optimization problem as

$$\hat{\rho} = \arg\min_{\rho \in \mathcal{C}_T} \mathbb{E}[L(\widetilde{\rho}, \rho) \mid y_{t_1}, \ldots, y_{t_T}] = \arg\min_{\rho \in \mathcal{C}_T} \sum_{\rho^* \in \mathcal{C}_T} L(\rho^*, \rho) P(\widetilde{\rho} = \rho^* \mid y_{t_1}, \ldots, y_{t_T}), \tag{2.14}$$

where $\hat{\rho}$ denotes the optimal point estimate of the latent random order, and $L(\cdot, \cdot) : \mathcal{C}_T \times \mathcal{C}_T \to \mathbb{R}$ denotes a loss function. Different choices can be made for the loss function, such as the $0-1$ loss function and the Binder loss function (Dahl, 2006; Vannucci and Do, 2007). Here we resort to the variation of information loss function (Meilă, 2007) to produce an optimal estimate of the latent order by considering the restricted support of the loss function. Even for small values of n, it is unfeasible to scan the entire order space C_n. Among the several solutions proposed in the literature, we restrict the optimization problem to sub-optimal solution within the space

$\mathcal{C}_T^R \subseteq \mathcal{C}_T$ of the orders visited in R steps of an MCMC sampling.

2.3 Simulation study

Through simulation studies, we investigate the capability of the model to recover the change points in sets of synthetic data by estimating the latent random orders. We consider four different scenarios in which we investigate the performance of the model with different specifications of the true distribution of the data. In each scenario, we generate synthetic data in which, between regimes, at least one of the following parameters of the data-generating distribution changes: the mean, the covariance matrix, or the dependency parameter. For instance, we consider a scenario in which only the mean varies across regimes, as well as a scenario in which all parameters change after a change point. This design allows us to assess the robustness of the model with respect to both the number and the type of parameters undergoing change. We expect, for example, the model to be less sensitive to changes in the covariance structure than to changes in the mean. For all scenarios, we simulate a three-dimensional time series $\boldsymbol{y}$, where the generic element takes values on $\mathbb{R}^3$ and $T = 300$. Furthermore we assume that the time series is composed by $j = 3$ regimes, induced by two change points, each one with a different parameter specification. With $y_{t_{j,i}}$ we denote the $i-$th element in regime j, where

$$y_{t_{j,i}} \mid y_{t_{j,i-1}}, \boldsymbol{\mu}_j, \Sigma_j \overset{\text{iid}}{\sim} \mathrm{N}(\gamma_j\, y_{t_{j,i-1}} + (1-\gamma_j)\boldsymbol{\mu}_j, \Sigma_j),$$

with the proviso $y_{t_{j,0}} = \boldsymbol{\mu}_j$. We set $m_1 = m_2 = m_3 = 100$. Different scenarios are generated according to the following specifications.

- Scenario 1: different means, same covariance matrix, and same dependency parameters. We set $\gamma_1 = \gamma_2 = \gamma_3 = 0.5$, $\boldsymbol{\mu}_1 = (-2,-2,-2)^\top$, $\boldsymbol{\mu}_2 = (3,3,3)^\top$, $\boldsymbol{\mu}_3 = (-1,-1,-1)^\top$ and

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}.$$

For Scenario 2,3 and 4 we preserve the same covariance structure

$$\Sigma_1 = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.2 & 0.5 & 0 \\ 0.3 & 0 & 0.5 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}, \quad \Sigma_3 = \begin{bmatrix} 0.35 & 0 & 0 \\ 0 & 0.35 & 0.1 \\ 0 & 0.1 & 0.35 \end{bmatrix}, \tag{2.15}$$

but with different values for the other parameters. Specifically we have

- Scenario 2: same means, different covariance matrices and same dependency parameters. We set $\gamma_1 = \gamma_2 = \gamma_3 = 0.5$, and $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \boldsymbol{\mu}_3 = (0,0,0)^\top$. The covariance structure is given by (2.15).

- Scenario 3: different means, covariance matrices and same dependency parameters. We set $\gamma_1 = \gamma_2 = \gamma_3 = 0.5$, $\boldsymbol{\mu}_1 = (-1,-1,-1)^\top$, $\boldsymbol{\mu}_2 = (0,0,0)^\top$ and $\boldsymbol{\mu}_3 = (1,1,1)^\top$. The covariance structure is given by (2.15).

- Scenario 4: different means, covariance matrices, and dependency parameters. We set $\gamma_1 = 0.2$, $\gamma_2 = 0.5$, $\gamma_3 = 0.8$, $\boldsymbol{\mu}_1 = (-1, -1, -1)^\top$, $\boldsymbol{\mu}_2 = (0, 0, 0)^\top$ and $\boldsymbol{\mu}_3 = (1, 1, 1)^\top$. The covariance structure is given by (2.15).

We consider four different specification of the model: for the first specification we set $\boldsymbol{m}_0 = \mathbf{0}$, $S_0 = \text{diag}(1)$, $\kappa_0 = 0.25$ and $\nu_0 = 4$, where diag(1) denotes a diagonal matrix with entries equal to 1. The second specification is parameterized as $\boldsymbol{m}_0 = \mathbf{0}$, $S_0 = \text{diag}(1)$, $\kappa_0 = 0.25$, and $\nu_0 = 10$. For the third specification we have $\boldsymbol{m}_0 = \mathbf{0}$, $S_0 = \text{diag}(1)$, $\kappa_0 = 0.5$, and $\nu_0 = 4$. In the fourth specification we set $\boldsymbol{m}_0 = \mathbf{0}$, $S_0 = \text{diag}(1)$, $\kappa_0 = 1$, and $\nu_0 = 4$. We complete the model specification by setting prior distributions on the main parameters, with $\delta \sim \text{Gamma}(2, 0.2)$, $\sigma \sim \text{Unif}(0, 1)$, and $\gamma \sim \text{Unif}(0, 1)$. We ran the estimations for 6 000 iterations, of which 5 000 were burn-in iterations. We randomly selected some of the replications and visually inspected the traceplots without any suggestion against the convergence of the algorithm. To assess the quality of the posterior estimates, we resort to two summaries of the posterior point estimates of the latent random orders $\hat{\rho} = (A_1, \ldots, A_{\hat{k}})$, namely, the number of distinct blocks k, and the normalized variation of information Meilă (2007) between the posterior point estimate and the true value of the latent random order $\rho_0 = (A_1, ..., A_{k_0})$, defined as

$$\text{VI}(\hat{\rho}, \rho_0) = \frac{1}{log(n)}(H(\hat{\rho}) + H(\rho_0) - 2I(\hat{\rho}, \rho_0)),$$

where $H(\rho) = \sum_{j=1}^{m} p_j(\rho) log(p_j(\rho)))$ represents the entropy associated to the partition ρ, while

$$I(\hat{\rho}, \rho_0) = \sum_{i=1}^{\hat{k}} \sum_{j=1}^{k_0} p_{ij}(\hat{\rho}, \rho_0) log[p_{ij}(\hat{\rho}, \rho_0)/(p_i(\hat{\rho}) p_j(\rho_0))]$$

denotes the mutual information of $\hat{\rho}$ and ρ_0, with the generic $p_j(\rho) = |A_j|/T$ and $p_{ij}(\hat{\rho}, \rho_0) = |A_i \cap A_0|/T$, where k, $\hat{k}$, k_0 denote the cardinality of ρ, $\hat{\rho}$, ρ_0 respectively. Table 2.1 shows the posterior summaries of the simulation study. We can see that the performance of the model is acceptable for all the scenarios considered. The posterior estimate of the latent random order of the data matches the true data-generating process in most cases. Within the scenarios with changes in the covariance structure, but not in the mean, the posterior point estimates of the latent random orders tend to be close to the true change points, but with some noise. Furthermore, the posterior point estimates of the dependence parameter γ are close to the true value. We are confident that the model is capable of producing good estimates of the latent order in the data. We further compared the posterior estimates of model (2.13) with two of its main competitors: the nonparametric approach introduced by Matteson and James (2014) and the Bayesian change point analysis method of Barry and Hartigan (1993).

Table A.1 shows the main summaries of the comparison. The nonparametric approach of Matteson and James (2014) shows overall better performances than the Bayesian approach of Barry and Hartigan (1993), but the latent orders estimated with model (2.13) have lower VI distance values from the true order, for all the scenarios considered in the study, while model (2.13) shows similar performances to the nonparametric approach of Matteson and James (2014) in terms of number of blocks.

Scenario	Model specification	$\hat{\vartheta}$	$\hat{\sigma}$	$\hat{\gamma}$	$\hat{k}$	VI$(\hat{\rho}, \rho_0)$
1	Spec. 1 – $(0, 0.25, 4, \text{diag}(1))$	0.616	0.091	0.491	3	0
	Spec. 2 – $(0, 0.25, 10, \text{diag}(1))$	0.577	0.087	0.482	3.16	0.001
	Spec. 3 – $(0, 0.5, 4, \text{diag}(1))$	0.603	0.091	0.495	3.06	0.001
	Spec. 4 – $(0, 1, 4, \text{diag}(1))$	0.620	0.0971	0.498	3.46	0.002
2	Spec. 1 – $(0, 0.25, 4, \text{diag}(1))$	0.907	0.122	0.492	4.04	0.104
	Spec. 2 – $(0, 0.25, 10, \text{diag}(1))$	1.082	0.105	0.476	4.54	0.121
	Spec. 3 – $(0, 0.5, 4, \text{diag}(1))$	1.099	0.308	0.490	5.3	0.110
	Spec. 4 – $(0, 1, 4, \text{diag}(1))$	1.310	0.204	0.493	8.2	0.128
3	Spec. 1 – $(0, 0.25, 4, \text{diag}(1))$	0.629	0.093	0.492	3.12	0.001
	Spec. 2 – $(0, 0.25, 10, \text{diag}(1))$	0.755	0.091	0.479	3.44	0.008
	Spec. 3 – $(0, 0.5, 4, \text{diag}(1))$	0.657	0.094	0.492	3.24	0.003
	Spec. 4 – $(0, 1, 4, \text{diag}(1))$	0.641	0.093	0.494	3.2	0.001
4	Spec. 1 – $(0, 0.25, 4, \text{diag}(1))$	1.057	0.100	0.499	5.14	0.038
	Spec. 2 – $(0, 0.25, 10, \text{diag}(1))$	1.150	0.095	0.491	5.92	0.042
	Spec. 3 – $(0, 0.5, 4, \text{diag}(1))$	1.162	0.102	0.496	6.04	0.055
	Spec. 4 – $(0, 1, 4, \text{diag}(1))$	0.875	0.098	0.511	4.26	0.025

Table 2.1: Posterior summaries for the simulation study. For each scenario and for each specification we report the posterior point estimate of δ, σ and γ, the number of blocks in the point estimate of the latent order in the data, and the variation of information distance between the point estimate of the latent random order and the true one. The results are averaged over 50 replications.

2.4 Application

We consider three distinct applications of the model. The first one to detect change points in the composition over time of the daily cases of COVID-19 across different areas of Italy, the second one to find change points on the annual temperature curves in Milan (Italy) and the third one to detect structural changes in the inflation rates of 12 key categories of goods in the European Union.

2.4.1 Change point detection in the COVID-19 Italian compositional data

We analyze a set of measurements about the incidence of COVID-19, as new daily cases, measured in each Italian macro-area. Data are collected by Protezione Civile, and they are available at https://github.com/pcm-dpc/COVID-19. We denote by the generic M_{ij} the number of new cases of the $i-$th day in the $j-$th area of Italy, where distinct macro-areas are represented by "north", "center", and "south and islands". Data were observed for a time window ranging from 2020/02/20 to 2021/02/23, for a total of $n = 364$ measurements. At the time of data analysis, the observations for 2020/02/20,2020/02/21 and 2020/06/09 were missing. We constructed the daily composition across the "north", "center", and "south and islands" areas by defining a vector $\boldsymbol{D}_i = (D_{i1}, D_{i2}, D_{i3})$ such that:

$$D_{ij} = \frac{M_{ij}}{\sum_{l=1}^{3} M_{il}}, \quad i = 1, \dots, n, \quad j = 1, 2, 3,$$

where $\boldsymbol{D}_i$ takes values on the two-dimensional simplex $\Delta_2 = \{(w_1, w_2, w_3) \in \mathbb{R}^3_+ : \sum_{j=1}^3 w_j = 1\}$. We then applied the additive log-ratio transform (Aitchison, 1982) to $\boldsymbol{D}_i$ for $i = 1, \ldots, n$, mapping the compositions to a real space through $g : \Delta_2 \to \mathbb{R}^2$ with:

$$\boldsymbol{X}_i = g(\boldsymbol{D}_i) = \left(\log\left(\frac{D_{i1}}{D_{i3}}\right), \log\left(\frac{D_{i2}}{D_{i3}}\right) \right), \quad i = 1, \ldots, n,$$

to obtain a multivariate real-valued time series $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$. We use the model in Equation (2.13) to perform multiple change point detection on the composition of COVID-19 across different Italian macro-areas. We use dimension $D_{3,i}$ as a reference dimension of the proportion of daily cases in southern Italy, while $D_{1,i}$ and $D_{2,i}$ denote the proportions of daily cases in northern and central Italy respectively. We complete the model specification by setting a priori $\delta \sim \text{Gamma}(2, 0.2)$, $\sigma \sim \text{Unif}(0, 1)$, $\gamma \sim \text{Unif}(0, 1)$ and P_0 distributed as a $\text{NIW}(\boldsymbol{m}_0, \kappa_0, \nu_0, S_0)$ with parameters $\boldsymbol{m}_0 = \boldsymbol{0}$, $\kappa_0 = 0.25$, $\nu_0 = 4$ and $S_0 = \text{diag}(1, 0.20)$. We ran the estimation for 60 000 iterations of which 20 000 were burn-in iterations. Visual inspection of the traceplots suggested moderately good mixing and did not highlight any convergence issues.

Figure 2.1 shows the time series of the log-ratio transform of the COVID-19 composition as a function of time, emphasizing the posterior estimate of the change points, and the posterior probability of having a change point at each observed time. The posterior probabilities of the change points are shrunk towards the posterior estimate for most of the detected change points, except for the last change point which shows a lager posterior uncertainty.

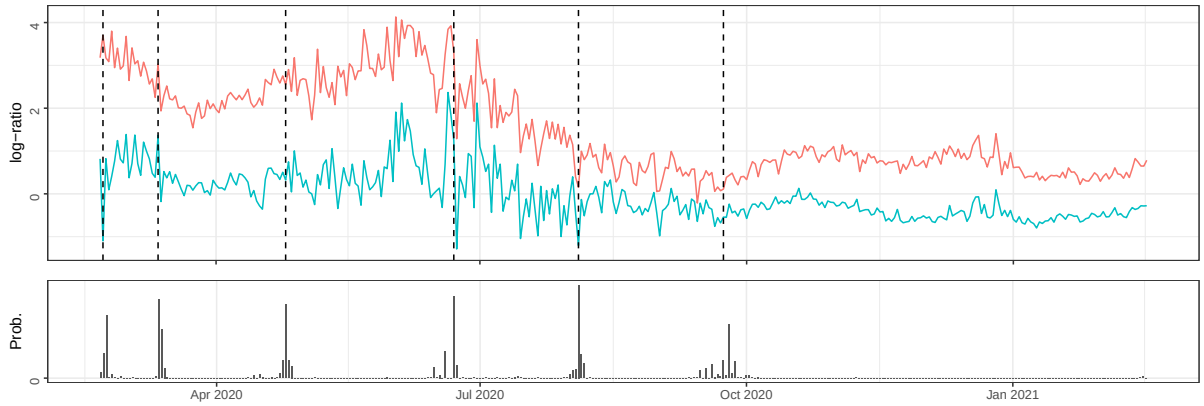


Figure 2.1: Posterior summaries of the model for the composition of new daily cases of COVID-19 in Italy. Top panel: time series of the log-ratio transform of the daily composition, the vertical dashed lines denote the posterior estimate of the change points. The red and the blue lines denote respectively the northern and the central part of Italy, both using the "south and islands" as reference dimension. Bottom panel: posterior probability of having a change point at each observed time.

By using the estimated change points, we can discuss the heterogeneity of the virus spreading across different areas. We first notice that the log-ratio transform is close to zero when both the dimension and the reference dimension share similar composition values. Thus, we can appreciate how the northern side of Italy, emphasized in red in Figure 2.1, captures most of the mass in the composition of the COVID-19 daily new cases in the period ranging from the start of the pandemic to summer. Within this period, we estimate three change points: a non-

informative change point in the early part of the time series, a second change point close to the introduction of the first restrictions in Italy, and a third change point that occurs close to the discharge of the first restrictions. During the summer period, we can see that the cases in the central part of Italy, denoted by the blue line in Figure 2.1, tend to show some spikes with respect to the reference dimension. Furthermore, during the summer, the line denoting the northern side of Italy drops to lower values, meaning that the proportion of cases becomes closer island area. We then find two more change points, one in the last part of the summer, where the time series starts to be stable but with some variability, and one right before the introduction of new restriction rules in Italy. After the last change point the time series appears stable, with an overall higher value of the proportion of cases in the northern part of Italy, with respect to the south and islands. We can appreciate how the second phase of limitations, which differs by region based on the recent spread of the virus, is more heterogeneous across different areas of Italy, and from the last change point detected, the composition appears stable over the observed time. The large number of change points on the left side of the curves suggest that during the first wave the spreading of the virus has been heterogeneous, with a major spread in northern Italy. The absence of change points on the right tail of the curves suggests that the second wave has affected the different Italian areas proportionally.

2.4.2 Analysis of the annual temperature curves in Milan data

We consider a set of daily measurements of the temperature detected in Milan, observed from January 1, 1763 to December 31, 2008. Figure 2.2 shows the realisations for a restricted time window, ranging from January 1, 1765 to December 31, 1771. We are interested in finding structural changes in the pattern of the temperature behaviours. With this aim, we consider the measurements grouped by year as realisations of a functional time series. We then have $X_1(s), \ldots, X_T(s)$ be realisations of random functions, where the generic $X_i(s)$ takes values on a functional space $\mathbb{F}$, endowed with its cylinder $\sigma-$field $\mathcal{F}$. For the generic observation time i, corresponding to a specific year, we observe an entire random function $X_i(s)$ indexed by a within-function time parameter s.

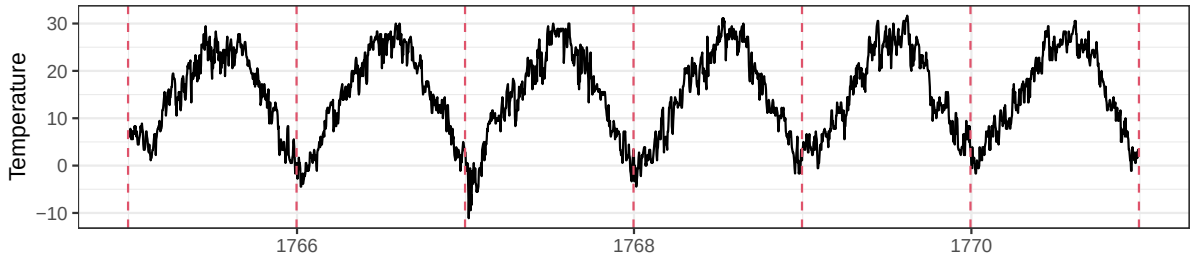


Figure 2.2: A sample of the observed time series for the period ranging from 1 January 1765 to 31 December 1771. Measurements are expressed in Celsius degrees. The red dashed vertical lines denote different years.

Except for some local noise, we assume that the behaviours of the curves is sufficiently regular to exhaustively represent the observations through a basis expansion. We then consider the projection of the observed curves on a low-dimensional space, where the generic $i-$th element

can be represented as

$$X_i(s) = \sum_{l=1}^{L} \beta_{i,l} b_l(s) + \epsilon_i(s), \quad i = 1, \dots, T, \tag{2.16}$$

with $\{b_1(s), \dots, b_L(s)\}$ representing a suitable set of function basis, and the generic β_i takes values on $\mathbb{R}^L$. Ignoring the nuisance term $\epsilon_i(s)$, we apply a change point detection model, as in Equation 2.13, directly to the coefficients $\beta_1, \dots, \beta_T$. We consider $K = 5$ distinct cubic B-spline functions as the basis system, with equally spaced knots. We complete the model specification by setting as prior distribution $\delta \sim \text{Gamma}(2, 0.2)$, $\sigma \sim \text{Unif}(0, 1)$, $\gamma \sim \text{Unif}(0, 1)$, and P_0 distributed as a $\text{NIW}(\boldsymbol{m}_0, \kappa_0, \nu_0, S_0)$ with parameters $\boldsymbol{m}_0 = \mathbf{0}$, $\kappa_0 = 1$, $\nu_0 = 1$ and $S_0 = \text{diag}(30, 80, 65, 65, 30)$, where S_0 is similar to the covariance of the basis expansion coefficients $\boldsymbol{\beta}_i$'s in Equation 2.16. We ran the estimation for 60 000 iterations, of which 20 000 were burn-in iterations. We further visually investigate the convergence of the chains, without any findings on it.

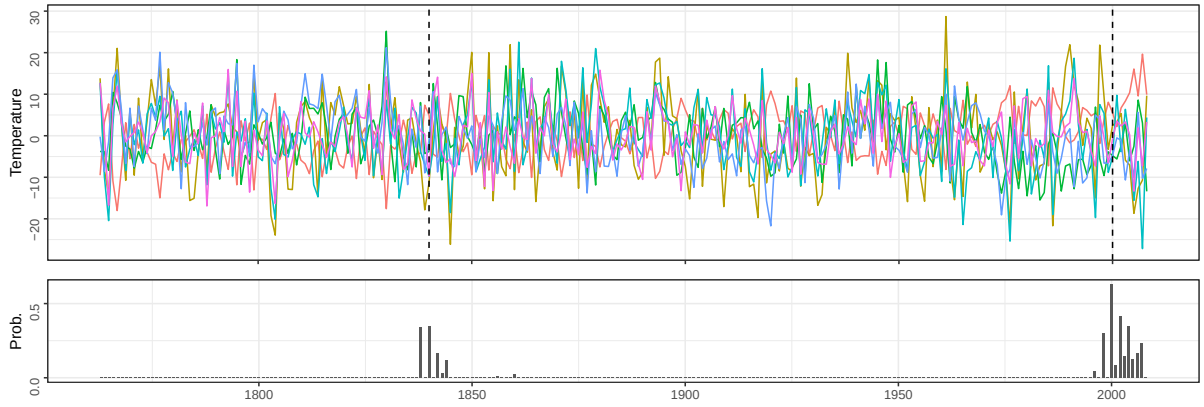


Figure 2.3: Posterior summaries of the model for the projection of the annual temperature curves in Milan. Top panel: time series of the projection of the curves on a spline basis system, with in evidence the posterior estimate of the change points. Bottom panel: posterior probability of having a change point at each observed time.

Figure 2.3 shows the posterior estimates of the change points and their distributions over the observed time interval. The model detects two potential change points. The posterior point estimate of the change points leads to the first occurrence in 1841, which historically corresponds to the period immediately after the first industrial revolution. A second change point is estimated in 2001, with a larger uncertainty associated with the point estimate, as shown in the bottom panel of Figure 2.3. This last change point falls within a time window which is recognized as the period immediately after the introduction of restrictions to mitigate global warming.

2.4.3 Change points in EU inflation rates

Inflation measures the overall rise in prices. When prices increase, the values of currencies decline and general goods are more expensive. While a restrained level of inflation is a sign of healthy economy, on the other hand with large value of inflation people spend more for primary goods, like food and energy. In recent years, inflation increased drastically in Europe due to many factors. One major reason is the imbalance between supply and demand caused by the

shortages and delays in the production after COVID-19 pandemic. Another key factor is the rise of energy prices, following the Ukraine invasion, after which many European countries needed to quickly seek alternative sources of energy. To analyze inflation, Eurostat, the statistical office of European Union, divides goods in 12 primary categories according to the classification of individual consumption by purpose (COICOP). For each category, a representative group of items is selected. The prices are then monitored throughout time. The most common index to measure inflation is the inflation rate that measures how prices for each category changed from the previous month or from the same month of the previous year. In this study, we want to detect structural changes that happened jointly in the inflation rates from 1997 to 2024 across the COICOP categories. Since the European Union underwent changes during the analyzed period— with some countries joining and the UK leaving in 2020— the inflation rate for each year is calculated based only on the countries that were part of the EU at that time.

For our analysis, we consider the annual average rate of change in the Harmonized Index of Consumer Prices (HICP) across the 12 categories of the COICOP classification from 1997 to 2024, resulting in a total of 336 observations for each category. In Table 2.2 are reported the categories with the corresponding code and label, data are available in the Eurostat database (https://ec.europa.eu/eurostat/data/database).

Code	Label
CP01	Food and non-alcoholic beverages
CP02	Alcoholic beverages, tobacco and narcotics
CP03	Clothing and footwear
CP04	Housing, water, electricity, gas and other fuels
CP05	Furnishings, household equipment and routine household maintenance
CP06	Health
CP07	Transport
CP08	Communications
CP09	Recreation and culture
CP10	Education
CP11	Restaurants and hotels
CP12	Miscellaneous goods and services

Table 2.2: Codes and labels of COICOP classification

We run the algorithm for 10 000 iterations, we set $k_0 = 0.5$, $\nu_0 = 20$ and $S_0 = \text{diag}(0.1, \dots, 0.1)$. We discard 2 500 iterations as burn-in period and to compute the final change points we use the search algorithm called SALSO presented in Dahl et al. (2022). This method is a very recent and more efficient alternative to all the classical methods for obtaining a point estimate in Bayesian clustering by minimization of a loss function. In this application, the posterior estimate under the variation of information loss led to a large number of change points and short regimes, resulting in a difficult interpretation of the final result. For this reason, we instead run the SALSO algorithm using the Binder loss function, which tends to produce fewer regimes, leading to a better interpretation of the estimated change points. We set 50 number of runs and 25 number of zealous updates. In Figure 2.4 are shown the estimated change points and the frequency of times that a change point has been selected in the MCMC chain. The latter plot suggests

that the final estimate is quite robust since we do not have a large variability in the posterior sample. The algorithm selects 11 change points. The first one is in December 1999 and is de-

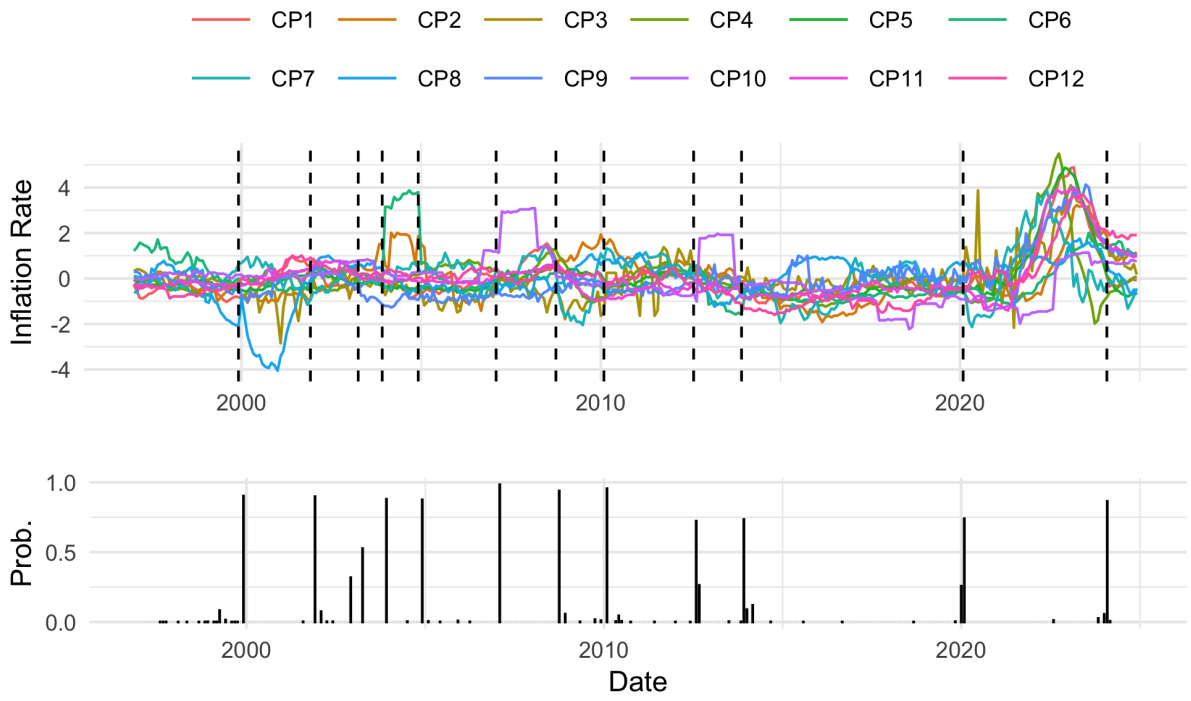


Figure 2.4: Posterior summaries of the model for the annual rates of change in the HICP index of the 12 COICOP categories. Top panel: time series of the inflation rates; colours denote the COICOP classification and the dashed lines the change points. Bottom panel: posterior probability of having a change point at each observed time.

termined by the telecoms crash that happened in the beginning of the new century, the second one is in December 2001 when prices stabilized after this crash and the third one in December 2002. A fourth change point occurs at the end of 2003 and it corresponds to the increasing of prices in the healthcare sector and in the alcoholic beverages, tobacco and narcotics products. During August 2006 another change point occurs where prices related to education increased rapidly. This change point is probably related to a federal reform that occurred in that period in Germany, where many local states introduced tuition fees for universities. The end of this period is denoted by a change point in October 2008. The following change point, that does not suggest any interpretations, is detected in February 2010. Then we have a period between two changes, one in August 2012 and one in December 2013, where again the prices related to education increased significantly. In this period, because of the austerity measures introduced by the main governments in Europe, tuition fees increased significantly in the UK in Spain and in Italy. Finally the two remaining change points that occurs in February 2020 and February 2024 are related with the consequences of COVID-19 pandemic and the invasion of Ukraine. In this period prices related to all sectors increased drastically.

2.5 Discussion

We extended one of the main state-of-the-art models to perform multiple change point detection within time series data to the multivariate case by accommodating a kernel for multivariate and possibly continuous time realisations. We derived the main quantities needed to perform posterior inference efficiently, particularly the integrated marginal likelihood. We further explicitly express the distribution of a single element given the rest of the data, which can be exploited to perform data imputation and estimate the model in the presence of missing observations. The model shows good performances on simulated data, even when we modify only the covariance structure across different blocks of the time series. We then applied the model to three real data scenarios, identifying change points in the composition of COVID-19 across different Italian macro-areas, in the annual temperature curves measured in the city of Milan and the inflation rate across European Union. Our findings support the well-established historical interpretations of the phenomena under study. We further note that the derivation of Equation 2.3, starting from the EPPF of a Pitman-Yor process, remains valid for more general classes of EPPFs, such as the one arising from a Gibbs-type prior. Nevertheless, an EPPF of the Gibbs-type form preserves the same effect of the discount parameter (see e.g. Lijoi et al., 2007), but possibly increases the computational complexity of the model. Furthermore, the model can be relaxed by considering a block-specific dependence parameter γ, capturing different dependencies across the time series.

Chapter 3

Model-Based Clustering of Time Dependent Data with Common Change Points

3.1 Introduction

Detecting patterns and similarities between observations is a key topic in modern statistics, especially in large datasets from which we can summarise information. Among fundamental approaches, cluster analysis comprises a collection of statistical tools to gather insights from a set of data by looking for groups with similar features, defining some homogeneous subsets of observations. In particular, in a model-based clustering setting, data are described by a mixture model where two observations belong to the same cluster if they are assigned to the same component of the mixture. Equivalently, each observed datum is associated with a latent parameter, which indexes a specific mixture component, and two observations are clustered together if their latent parameters have the same value. Here we consider a scenario involving multiple time series, with each observation representing a series of values observed over time. We introduce a new method for clustering these time series based on specific characteristics, such as shared structural changes over time.

We introduce an approach that employs a model-based clustering framework to group observations exhibiting common structural changes, while other individual parameters and regimes are observation-specific. Our goal is to cluster observations that experience a shock simultaneously, even if their behaviours differ. To our knowledge, the use of latent combinatorial structures for model-based clustering of time series, that primarily focus on change points, has not been investigated yet. However, there are various contributions aimed at clustering multiple time series, possibly with regime changes. For a comprehensive review of different clustering techniques for time series see Aghabozorgi et al. (2015). In a Bayesian framework, early contributions to model-based clustering of time series can be found in Frühwirth-Schnatter and Kaufmann (2008), who examined a scenario with a fixed number of clusters, grouping observations based on latent parameters that define their behaviours. Later extensions include different kernel functions (Juárez and Steel, 2010) and categorical data (Frühwirth-Schnatter and Pamminger,

2010), among others. Allied to our approach, Same et al. (2011) propose a model to cluster together time series with changes in their regime, by resorting to a polynomial representation of the time dependent observations, whose degree can be different over time. Further, Brault et al. (2024) introduce a method that clusters together functional data with structural changes. The latter not only clusters data with similar behaviours, but also detects change points for the functions in each group. However, differently from our approach, here the clustering is focused on similarities of local behaviours, rather than primarily on change points.

Our primary motivation behind this study comes from the recent COVID-19 pandemic. Epidemic models such as the Kermack–McKendrick renewal equation models (Kermack and McKendrick, 1927) and their stochastic analogues have been studied for many decades now (Andersson and Britton, 2000). Along with their study, parameter inference methods have been also developed (see, e.g, Niels G., 1993; Becker, 1993; Choi and Rempala, 2011; Raúl Fierro and Balakrishnan, 2015; Kypraios et al., 2017). See Britton and Pardoux (2019) for an overview on this topic. Due to the recent COVID-19 pandemic, these methods received a renewed interest. For example, frequentist methods (Bong et al., 2024) and other techniques for calibration of computer models (Fadikar et al., 2018; Baker et al., 2022) have been investigated successfully in the last years. Similarly, the problem of parameter inference from a Bayesian perspective was discussed for mechanistic (Seymour et al., 2022; Chitwood et al., 2022; KhudaBukhsh et al., 2023) and semi-mechanistic models of COVID-19 (Bhatt et al., 2023). More recent contributions accommodate for different behaviour of the model over time. Specifically, Hong and Li (2020), Gleeson et al. (2021) and Wascher et al. (2024) discussed models with time-varying infection rates, while structural changes in such parameters were modeled in Huang et al. (2024). Here, we want to cluster different countries on the basis of their structural changes in the dynamics dictated by a stochastic epidemic model.

The chapter is structured as follows. Section 3.2 introduces a model to cluster together time-dependent observations with common change points. Section 3.3 describes the distributions of interest that can be used to construct a sampling strategy. In Section 3.4 we present a Markov chain Monte Carlo (MCMC) procedure to sample realisations from the posterior distribution of interest. In Section 3.5 the properties of the model are studied with an application to synthetic and real time series data. Section 3.6 shows the epidemiological study which motivates our contribution. Conclusions are in Section 3.7. Further details on algorithm implementations, illustrations, and the epidemiological model are deferred to Appendix B.

3.2 Modeling multiple time series with common change points

We introduce a novel approach to cluster together time series with respect to common structural changes, and no further local similarities are assumed among them. In this sense, observations might be locally indexed by different subject-specific behaviours, but cluster together if they change regime on the same time instants. Let $\boldsymbol{y}_i = (y_{i,1}, \dots, y_{i,T})$, be the generic ith time series we observe, where $y_{i,t}$ is a random quantity taking values on a space $\mathbb{Y}$, which depends on the specific quantity we observe, and $\mathcal{Y} = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_n\}$ denotes the whole sample. For simplicity, here we assume that time series are observed at the same discrete times $(1, \dots, T)$, however our

approach works even in more general scenarios. See, e.g., Section 3.6 where we consider a model tailored for epidemiological spreads as kernel function.

Regarding the distribution of each time series, we assume that our data are sharing the same model structure over time with Markovian dependence, where $\mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta_{i,t})$, $t = 1, \dots, T$ denotes the distributional law of $y_{i,t}$ given the previous observation $y_{i,t-1}$ and the local parameter $\theta_{i,t}$, with $\theta_{i,t} \in \Theta$ for all $i = 1, \dots, n$ and $t = 1, \dots, T$. We remark that the modeling structure presented in the manuscript can be extended to cases with longer memories by substituting the distribution $\mathcal{L}$ with suitable alternatives. The local behaviours of the model for the ith observation are dictated by the sequence of latent parameters $\theta_{i,1}, \dots, \theta_{i,T}$, which are assumed to have ties over time. A change point for the ith time series occurs when the parameter at time t differs from the previous value, i.e. $\theta_{i,t} \neq \theta_{i,t-1}$. Equivalently, each observation is divided into one or more blocks, where realisations within each block share the same value of $\theta_{i,t}$. When the value of the parameter changes a change point occurs and a new block begins. The distribution of the sequence of parameters $\theta_{i,1}, \dots, \theta_{i,T}$ factorises into two independent terms

$$\mathcal{L}(\theta_{i,1}, \dots, \theta_{i,T}) = \mathcal{L}(\rho_i)\mathcal{L}(\theta^*_{i,1}, \dots, \theta^*_{i,k_i}), \tag{3.1}$$

as done in Section 2.2. The first term, $\mathcal{L}(\rho_i)$, denotes the distribution of the random order ρ_i, which induces changes in the sequence of parameters $\theta_{i,1}, \dots, \theta_{i,T}$. Here, by a random order ρ_i, we mean a random partition of $1, \dots, T$ into k_i blocks $A_{i,1}, \dots, A_{i,k_i}$. The second term in (3.1) denotes the distribution of $\theta^*_{i,1}, \dots, \theta^*_{i,k_i}$, the unique values out of $\theta_{i,1}, \dots, \theta_{i,T}$, whereas the generic $\theta^*_{i,j}$ determines a common local behaviour and it is shared by all the observations whose times are allocated in the jth block of ρ_i. Hence, a time series switches its regime when it moves from a block to another of $\rho_i = (A_{i,1}, \dots, A_{i,k_i})$, while the local behaviours are fully characterized by $\theta^*_{i,1}, \dots, \theta^*_{i,k_i}$.

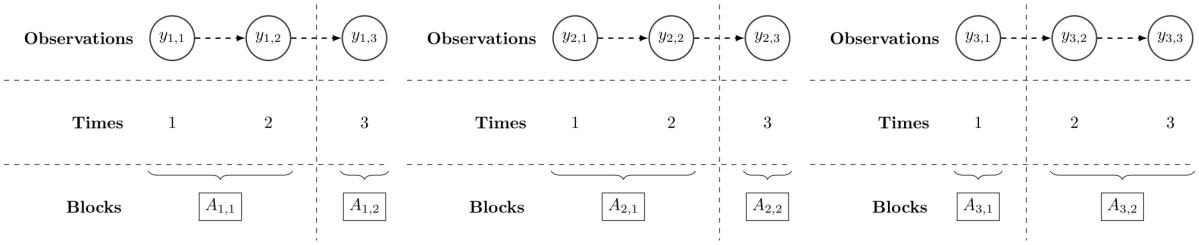


Figure 3.1: Graphical representation of three time series $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3$, each observed at $T = 3$ time points, with their corresponding latent orders. Here, $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ belong to the same cluster, while $\boldsymbol{y}_3$ is assigned to a separate cluster.

Our aim is to cluster time series in homogeneous groups sharing the same change points. To this end, we assume the latent orders $\mathcal{R} = \{\rho_1, \dots, \rho_n\}$ distributed independently according to a discrete distribution $\widetilde{p}$, with

$$\widetilde{p}(\rho) = \sum_{r=1}^{2^{T-1}} \pi_r \delta_{\widetilde{\rho}_r}(\rho), \tag{3.2}$$

where $\{\widetilde{\rho}_1, \dots, \widetilde{\rho}_{2^{T-1}}\}$ is the collection of all the possible orders of T elements, $\pi_1, \dots, \pi_{2^{T-1}}$ are the probabilities taking values in the $(2^{T-1} - 1)$-dimensional simplex space, and $\delta_a(\cdot)$ denotes

a Dirac measure at a. The discreteness of (3.2) implies a positive probability of having ties in $\mathcal{R}$. Ties represent clusters of observations sharing the same change points, i.e. changing their local behaviours at the same time. The observations are then divided according to a partition $\lambda = \{B_1, \dots, B_k\}$ of the indices $\{1, \dots, n\}$, for which $B_\ell \cap B_j = \emptyset$ for all $\ell, j \in \{1, \dots, k\}$ with $\ell \neq j$, and $B_1 \cup \cdots \cup B_k = \{1, \dots, n\}$. Figure 3.1 shows an example with $n = 3$ time series observed for $T = 3$ time instants with a single change point. The left and middle series belong to the same cluster, as the change point occurs at $t = 3$, while the right one forms a stand-alone cluster with a change point at time $t = 2$.

The model specification is completed by setting a distribution for the weights $\pi_1, \dots, \pi_{2^{T-1}}$, in the following sections assumed to be Dirichlet distributed, by choosing a distribution for the observed data $\mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta_{i,t})$ and by setting a prior distribution for the parameters $\theta_{i,t}$s, here denoted by P_0. We remark that, since each observed time series is associated with a latent order, its likelihood contribution can be factorised into the product of terms sharing the same local behaviour, as described in Section 2.2. The model we consider can then be written in its hierarchical form as

$$\begin{aligned} \boldsymbol{y}_i \mid \rho_i, \boldsymbol{\theta}_i^* &\sim \prod_{j=1}^{k_i} \prod_{t=t_{i,j}^-}^{t_{i,j}^+} \mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta_{i,j}^*), \quad i = 1, \dots, n, \\ \rho_i \mid \widetilde{p}(\rho) &\overset{\text{iid}}{\sim} \widetilde{p}(\rho), \quad i = 1, \dots, n, \\ (\pi_1, \dots, \pi_{2^{T-1}}) &\sim \text{DIR}(\alpha_1, \dots, \alpha_{2^{T-1}}), \\ \theta_{i,j}^* &\overset{\text{iid}}{\sim} P_0(\theta), \quad j = 1, \dots, k_i, \ i = 1, \dots, n, \end{aligned} \tag{3.3}$$

where $t_{i,j}^- = \min\{t : t \in A_{i,j}\}$ denotes the first time index belonging the jth block of the latent order associated with the ith observation, $t_{i,j}^+ = \max\{t : t \in A_{i,j}\}$ the last time index, and with the proviso that the first time of a new regime in the observed quantities does not depend on observed values in past regimes, i.e. $\mathcal{L}(y_{i,t_{i,j}^-} \mid y_{i,t_{i,j}^- - 1}, \theta_{i,j}^*) = \mathcal{L}(y_{i,t_{i,j}^-} \mid \theta_{i,j}^*)$.

Specific distributional choices of $\mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta_{i,t})$ depend on the data we are considering. For example, in the following sections we present an illustration where the data are real-valued quantities modeled through an univariate Ornstein-Uhlenbeck process (Section 3.5) and a case with survival data arising from a susceptible-infected-removed model (Section 3.6). However, any time-dependent model with discrete or discretisable observational time can be embedded in the specification of a model as in (3.3). The distribution of the parameters $\mathcal{T}^* = \{\boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_n^*\}$ depends on the distributional assumption of $\mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta_{i,t})$. Futher, $\mathcal{T}^*$ can be marginalised in case we are not interested on posterior inference for local behaviours, but only on common structural change times.

3.3 Posterior distribution of interest

Our main interest lies in the latent partition defining groups of time series which share the same change points. The posterior distribution of such quantity is not available in a closed form, but we can identify a tractable expression from which we can produce a sample. Hence, we

start from the joint distribution of the observed time series $\mathcal{Y}$, latent orders $\mathcal{R}$, unique values of the parameters $\mathcal{T}^*$ and the vector of probabilities $\boldsymbol{\pi} = (\pi_1, \dots, \pi_{2^{T-1}})$. The distribution of the previous quantities can be then expressed, in force of the chain rule, as product of the corresponding conditional distributions, with

$$\mathcal{L}(\mathcal{Y}, \mathcal{T}^*, \mathcal{R}, \boldsymbol{\pi}) = \mathcal{L}(\mathcal{Y} \mid \mathcal{T}^*, \mathcal{R}) \mathcal{L}(\mathcal{T}^* \mid \mathcal{R}) \mathcal{L}(\mathcal{R} \mid \boldsymbol{\pi}) \mathcal{L}(\boldsymbol{\pi}), \tag{3.4}$$

where $\mathcal{Y}$ is independent of $\boldsymbol{\pi}$ conditioned on $\mathcal{T}^*$, $\mathcal{R}$ and $\mathcal{T}^*$ is also independent of $\boldsymbol{\pi}$ conditioned on $\mathcal{R}$. We can make the joint distribution in (3.4) explicit by using the distributions in (3.3), with the likelihood term, here denoted by $\mathcal{L}(\mathcal{Y} \mid \mathcal{T}^*, \mathcal{R})$, being the product of the data distribution for all the observed time series. We then have

$$\begin{aligned} \mathcal{L}(\mathcal{Y}, \mathcal{T}^*, \mathcal{R}, \boldsymbol{\pi}) = & \prod_{i=1}^{n} \prod_{j=1}^{k_i} \prod_{t=t_{i,j}^-}^{t_{i,j}^+} \mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta_{i,j}^*) \\ & \times \prod_{i=1}^{n} \prod_{j=1}^{k_i} P_0(\theta_{i,j}^*) \prod_{i=1}^{n} \widetilde{p}(\rho_i) \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{r=1}^{2^{T-1}} \pi_r^{\alpha_r - 1}, \end{aligned} \tag{3.5}$$

where $B(\boldsymbol{\alpha})$ denotes the multivariate beta function of parameter $\boldsymbol{\alpha}$.

We are interested in partitioning the observations on the base of their latent orders. Therefore, the unique values of the parameters $\mathcal{T}^*$ can be marginalised from the previous equation, obtaining the following expression

$$\mathcal{L}(\mathcal{Y}, \mathcal{R}, \boldsymbol{\pi}) = \prod_{i=1}^{n} \prod_{j=1}^{k_i} \mathcal{M}(\{y_{i,t} \mid t \in A_{i,j}\}) \prod_{i=1}^{n} \sum_{r=1}^{2^{T-1}} \pi_r \delta_{\widetilde{\rho}_r}(\rho_i) \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{r=1}^{2^{T-1}} \pi_r^{\alpha_r - 1},$$

whereas $\mathcal{M}(\{y_{i,t} \mid t \in A_{i,j}\})$ denotes the marginal distribution of the ith time series in the jth block, resulting from integrating the local parameter $\theta_{i,j}^*$, i.e.

$$\mathcal{M}(\{y_{i,t} \mid t \in A_{i,j}\}) = \int \prod_{t=t_{i,j}^-}^{t_{i,j}^+} \mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta_{i,j}^*) P_0(\theta_{i,j}^*) \mathrm{d}\theta_{i,j}^*. \tag{3.6}$$

Clearly, the tractability of (3.6) depends on specific choices of the data distribution $\mathcal{L}(y \mid \cdots)$ and the prior for the local parameter $Q_0(\theta_{i,j}^*)$. In some common scenarios, the integral can be solved analytically, obtaining a closed form expression for the marginal distribution, as done for example in Section 3.5. Alternatively, for more complex scenarios such integral can be approximated through numerical solutions. See, e.g., Section 3.6.

Finally, we can marginalise also the weights $\boldsymbol{\pi}$, as we are not interested on estimates of the mixing distribution but on the clustering, obtaining the following distribution

$$\mathcal{L}(\mathcal{Y}, \mathcal{R}) = \frac{\Gamma(\alpha^+)}{\Gamma(\alpha^+ + n)} \prod_{r=1}^{2^{T-1}} \frac{\Gamma(\alpha_r + n_r)}{\Gamma(\alpha_r)} \prod_{\{i:\rho_i = \widetilde{\rho}_r\}} \prod_{j=1}^{k_i} \mathcal{M}(\{y_{i,t} \mid t \in A_{i,j}\}), \tag{3.7}$$

where $\alpha^+ = \sum_{r=1}^{2^{T-1}} \alpha_r$, $\{i : \rho_i = \widetilde{\rho}_r\}$ denotes the set of indices for which the latent order ρ_i coincides with the rth element of $\widetilde{p}$, and n_r is the cardinality of $\{i : \rho_i = \widetilde{\rho}_r\}$, i.e., number of observations associated with the cluster with latent order $\widetilde{\rho}_r$. A model of the form in (3.7) describes the convolution of a Dirichlet-categorical model with the marginal distributions of the data. We further remark that the partition of the data λ is then given by the equivalence classes described by allocating observations to different latent orders, with $\lambda = \{B_1, \ldots, B_k\}$ whereas $B_j = \{i : \rho_i = \rho^\dagger_{(j)}\}$ is the jth block of the partitions whose latent orders are equal to $\rho^\dagger_{(j)}$, and $\{\rho^\dagger_{(1)}, \ldots, \rho^\dagger_{(k)}\}$ denotes the set of orders out of $\widetilde{\rho}_1, \ldots, \widetilde{\rho}_{2^{T-1}}$ with at least one observation assigned, i.e. the elements in $\{\rho_r : n_r > 0\}$. In the following, the model is elicited by considering an objective prior specification for the weights, i.e., $\alpha_r = \alpha$ for all $r = 1, \ldots, 2^{T-1}$. We recall that the probability of having ties while sampling two observations from a Dirichlet-categorical model is given by

$$P(X_2 = X_1) = \frac{\alpha + 1}{2^{T-1}\alpha + 1}$$

with $\lim_{\alpha \to 0} P(X_2 = X_1) = 1$ and $\lim_{\alpha \to +\infty} P(X_2 = X_1) = \frac{1}{2^{T-1}}$. Therefore, to ensure a significant probability of a tie when dealing with large values of T, in the following we set $\alpha = 1$.

3.4 Informed split and merge algorithm

The model we consider in the manuscript has a flexible structure which can accommodate for various types of data. The bottleneck of facing posterior inference is due to two main sources: the distribution we assume for the data $\mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta_{i,t})$, which depends on specific analyses we are performing, and the cardinality of $\widetilde{p}$. Regarding the latter, the number of latent orders scales poorly with the number of observed values for the time series, with 2^{T-1} possible orders when we observe T distinct times. Even for a small T, we cannot explore the whole partitions' space in a reasonable time, and we need to use suitable computational strategies to obtain a sample from the target distribution of interest.

Among the possible approaches, we build a sampling strategy upon the split and merge algorithm for nonparametric priors and clustering problems (Green and Richardson, 2001; Jain and Neal, 2004), a particular type of MCMC algorithm that can be used to update the latent partition λ of a model as in (3.3). Intuitively, at each step of the algorithm two indices of the latent partition, $i, \ell \in \{1, \ldots, n\}$ with $i \neq \ell$, are randomly chosen. If the two indices belong to the same block of λ, meaning that the ith and the ℓth observations belong to the same cluster, their cluster is randomly split in two blocks. Otherwise, if the two indices i and ℓ belong to different blocks of λ, the two clusters are merged. Once the split or the merge step is completed, we obtain a candidate partition $\lambda^{(N)}$ to update the current state of chain. We then perform a Metropolis–Hastings step to accept $\lambda^{(N)}$ or to stay on the previous state of the chain. We report the algorithm to sample M values from the posterior distribution of the latent partition λ.

Main phase: during the main phase, the partition and the corresponding latent orders are updated by means of a split and merge procedure, which allows to explore locally the space of possible partitions. Let the initial partition be $\lambda^{(0)} = \{B_1^{(0)}, \ldots, B_k^{(0)}\}$ of $\{1, \ldots, n\}$, and denote by $\mathcal{R}^{*(0)}$ the initial values of the unique latent orders. For $m = 1, \ldots, M$, repeat the following

steps:

a) Set $\lambda^{(N)} = \lambda^{(m-1)}$, where $\{B_1^{(N)}, \dots, B_k^{(N)}\}$ denote the blocks of $\lambda^{(N)}$, and set $\mathcal{R}^{*(N)} = \mathcal{R}^{*(0)}$.

b) Sample two distinct indices $i, \ell \in \{1, \dots, n\}$.

- If both i and ℓ belong to the same block $B_s^{(m-1)}$, perform a *split*:
  - (i) assign i to $B_s^{(N)}$ and ℓ to a new block $B_{k+1}^{(N)}$;
  - (ii) assign randomly each element of $B_s^{(m-1)}$ either to $B_s^{(N)}$ or to $B_{k+1}^{(N)}$;
  - (iii) sample the two distinct latent orders in $\mathcal{R}^{*(N)}$ associated with the observations whose indices belong to $B_s^{(N)}$ and $B_{k+1}^{(N)}$, respectively, from (3.8).
- If i and ℓ belong to different blocks, with $i \in B_s^{(m-1)}$ and $\ell \in B_w^{(m-1)}$, perform a *merge*:
  - (iv) assign all indices in $B_w^{(m-1)}$ to $B_s^{(m-1)}$ and remove block $B_w^{(m-1)}$;
  - (v) sample the unique latent order in $\mathcal{R}^{*(N)}$ associated with all observations whose indices belong to $B_s^{(m-1)}$ from (3.8).

c) Perform a Metropolis–Hastings step to accept or reject the proposed values. If the proposal is accepted, set $(\lambda^{(m)}, \mathcal{R}^{*(m)}) = (\lambda^{(N)}, \mathcal{R}^{*(N)})$; otherwise, retain the previous configuration, that is $(\lambda^{(m)}, \mathcal{R}^{*(m)}) = (\lambda^{(m-1)}, \mathcal{R}^{*(m-1)})$.

d) Perform an acceleration step to update the unique latent values $\rho_1^{*(m)}, \dots, \rho_k^{*(m)}$ in $\mathcal{R}^{*(m)}$.

This iterative split and merge mechanism ensures a local but efficient exploration of the partition space while maintaining proper mixing through the Metropolis–Hastings acceptance rule.

The main challenge of using the split and merge algorithm in our framework is the sparse nature of $\widetilde{p}$. When we perform a split or merge step, we need to propose a new latent order for each new block we create, as in step iii) and v) of the algorithm. Because $\widetilde{p}$ has a large cardinality, randomly proposing those orders from a distribution independent of the data or previous state makes them unlikely to be representative of the new clusters, even when the proposed partition $\lambda^{(N)}$ is a suitable candidate. To avoid this problem, we resort to an informed proposal for those orders. In particular, inspired by Zhang et al. (2022), we included in the proposal specification the information arising from the observed time series. Hence, our proposal equals

$$\psi(\rho \mid \mathcal{Y}) = \sum_{i=1}^{n} \frac{1}{n} \mathcal{L}(\rho \mid \boldsymbol{y}_i), \tag{3.8}$$

i.e. a mixture of the posterior distribution of the latent orders conditionally on each observation separately. Here, $\mathcal{L}(\rho \mid \boldsymbol{y}_i)$ denotes the posterior distribution of the latent order ρ given the ith time series $\boldsymbol{y}_i$. Such mixture strategy for the proposal shrinks the probability mass across the space of orders $\rho_1, \dots, \rho_{2^{T-1}}$ to the elements which are likely at least for a single observations, decreasing the probability of sampling values which are not representative of any observed time series. While sampling from a generic component of (3.8) is feasible, evaluating

the entire mixture requires the computation of n normalisation constants, intractable for large T. These normalisation constants can be computed before starting the algorithm. For example, we considered an importance sampling strategy with uniform importance distribution over the partitions' space (see Section B.3 of the Appendix). Further details on the implementation are deferred to Section B.1 of the Appendix.

3.5 Real-valued time series clustering

Before extending our model strategy to more complicated scenarios, we first investigate the performance of the algorithm with a synthetic study on time series and then show an application on a real financial scenario. We consider a case where data are real-valued time series, for which structural changes occur at certain times. The generic jth block of the ith time series is generated according to the following process

$$y_{i,t} = \begin{cases} \mu_{i,t} + N(0, \eta_{i,t}), & \text{if } t = 1, \\ \gamma\, y_{i,t-1} + (1-\gamma)\, \mu_{i,t} + N(0, (1-\gamma^2)\eta_{i,t}), & \text{otherwise}, \end{cases} \tag{3.9}$$

where $N(a, b)$ denotes the Gaussian distribution with expectation a and variance b. We consider a scenario with three groups of time series. Each series is observed $T = 300$ times divided into k_i distinct regimes, $i = 1, \ldots, 10$, and series belonging to the same cluster have change points at the same time but different local parameters.

To assess the robustness of our model, we generate synthetic data under several scenarios in which, at each change point, one or both of the block-specific parameters of the data-generating distribution may vary, namely the mean and the variance. In particular, we consider scenarios in which only the mean changes across regimes, as well as more challenging settings in which only the variance changes. This design allows us to evaluate the robustness of the model with respect to both the number and the type of parameters undergoing change. Moreover, to evaluate the clustering capability of the proposed approach when the only shared characteristic across time series is the location of the change points, we allow time series within the same cluster to have distinct local parameters. For all time series, we fix the dependency parameter to $\gamma = 0.1$. Block-specific parameter values are defined by selecting unique values $\mu^*_{i,j}$ from $\mu_{i,1}, \ldots, \mu_{i,T}$ and $\eta^*_{i,j}$ from $\eta_{i,1}, \ldots, \eta_{i,T}$ according to the scheme reported in Table 3.1, so that change points occur at different time points across clusters. An example of the sampled data is shown in Figure B.1.

We analyze the data considering a model as in (3.3). In particular, we assume that within each block the data follow an univariate Ornstein-Uhlenbeck process (Uhlenbeck and Ornstein, 1930), similarly to Section 2.2. The process defined as the solution to the following stochastic differential equation

$$dY_t = -\alpha(Y_t - \mu)dt + \sqrt{\frac{2\alpha}{\eta}}dW_t, \tag{3.10}$$

where $\mu \in \mathbb{R}$, $\alpha, \eta > 0$, $\{W_t\}_{t \geq 0}$ is the Wiener process, and α is the parameter tuning the dependence over time. This specific model choice, once integrated at discrete times, leads to a model of the form in Equation 3.9. Here, the generic parameter indexing locally the time series is

i	$\{\mu^*_{i,1},\dots,\mu^*_{i,k_i}\}$	$\{\eta^*_{i,1},\dots,\eta^*_{i,k_i}\}$	$\{m_{i,1},\dots,m_{i,k_i}\}$
1	{-0.25,-0.5,0.5,0,0.5}	{0.1,0.12,0.14,0.1,0.13}	{50,100,45,55,50}
2	{0.1,0.12,0.14,0.1,0.13}	{0.1,0.12,0.14,0.1,0.13}	
3	{0.45,-0.5,0.5,0,0.5}	{0.1,0.12,0.2,0.12,0.14}	
4	{0.5,-0.5,0.5,0.5,0}	{0.1,0.12,0.09,0.24,0.15}	
5	{-0.5,0,1,-0.1,0.6,-0.2}	{0.14,0.13,0.17,0.12,0.14,0.12}	{40,50,45,45,30,90}
6	{0,0.85,-0.15,0.65,0,1}	{0.1,0.24,0.14,0.15,0.12,0.13}	
7	{0.4,0.4,-0.15,0.5,0,-0.65}	{0.22,0.12,0.1,0.13,0.14,0.12}	
8	{-0.5,0,1,-0.1,0.6,-0.2}	{0.14,0.13,0.17,0.12,0.14,0.12}	{75,50,40,20,75,40}
9	{0.5,0.5,1,0.25,-0.5,0.25}	{0.12,0.22,0.15,0.14,0.17,0.19}	
10	{-0.5,0,0.5,-0.1,0.6,-0.25}	{0.12,0.13,0.15,0.12,0.15,0.18}	

Table 3.1: Data generating process parameters for the the synthetic study with real-valued time series. Left to right: observation index, local trend parameters, local dispersion parameters, and true latent order shared among observations in the same cluster.

given by $\theta_{i,t} = (\mu_{i,t}, \eta_{i,t})$, while the parameter α (or equivalently $\gamma = e^{-\alpha}$) is assumed to be fixed and shared for all the time series. By setting $\mu \mid \eta \sim \text{Normal}(0, (c\eta)^{-1})$ and $\eta \sim \text{Gamma}(a, b)$, Martínez and Mena (2014) have shown that the marginal distribution of the data becomes

$$\begin{aligned}
&\mathcal{M}(\{y_{i,t} : t \in A_{i,j}\} \mid \gamma) \\
&\quad = \frac{(2b(1-\gamma^2))^a \Gamma(n_j/2 + a)}{\pi^{n_j/2}\Gamma(a)} \left(\frac{c(1+\gamma)(1-\gamma^2)}{c + n_j - \gamma(n_j - c - 2)}\right)^{1/2} \times \\
&\quad \left(\boldsymbol{y}_j^\intercal \boldsymbol{S}_j \boldsymbol{y}_j - \frac{(1-\gamma)(\sum_{i=1}^{n_j} y_{j,i} - \gamma \sum_{i=2}^{n_j - 1} y_{j,i})^2}{c + n_j - \gamma(n_j - c - 2)} + 2b(1-\gamma^2)\right)^{-(n_j/2 + a)},
\end{aligned} \tag{3.11}$$

where $\boldsymbol{y}_j = \left(y_{i,t^-_{i,j}}, \dots, y_{i,t^+_{i,j}}\right)$ and $\boldsymbol{S}_j$ is an $n_j \times n_j$ matrix of the form

$$\boldsymbol{S}_j = \begin{bmatrix} 1 & -\gamma & 0 & \dots & 0 \\ -\gamma & 1+\gamma^2 & -\gamma & \dots & 0 \\ 0 & -\gamma & 1+\gamma^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

Posterior inference is performed exploiting the algorithm presented in Section 3.4. We consider different scenarios by combining together different values of the parameters tuning such algorithm. Specifically, we vary the accuracy of the normalisation constant in Equation 3.8, by considering different numbers of values in the importance sampling $\text{B} \in \{1\,000, 10\,000, 100\,000\}$. Further, when we propose a candidate from Equation 3.8, we consider different depths of the proposal, i.e. the number of sampling steps $\text{L} \in \{1, 25, 100\}$. While proposing a candidate random order, we are randomly selecting one of the n dimensions in (3.8), then we sample from the corresponding posterior distribution. For the latter, we initialize randomly the latent order and we then perform L split and merge step (Martínez and Mena, 2014) to update that value. Each scenario considered in the study is replicated 50 times.

The model specification is completed by specifying the parameters of the Dirichlet distribution in Equation 3.7, with $\alpha_r = \alpha = 1$ for every $r \in \{1, \dots, 2^{T-1}\}$. Following the approach by Martínez and Mena (2014), the likelihood of a time series $\boldsymbol{y}_i$ is given by (3.11) where μ and η have been marginalised after setting respectively a normal and a gamma prior distributions. According to the notation introduced beforehand, we set as parameters for the gamma $a = b = 1$ and $c = 0.1$ for the normal distribution. Finally, we set the autocorrelation parameter $\gamma = 0.1$, matching the values used to simulate the data. For the split and merge procedures that compute the marginal change points we set the probability of performing a split to $q = 0.5$.

B	L	$\text{BI}(\hat{\lambda}, \lambda_0)$	B	L	$\text{BI}(\hat{\lambda}, \lambda_0)$	B	L	$\text{BI}(\hat{\lambda}, \lambda_0)$
	1	0.251		1	0.270		1	0.282
10^3	25	0.146	10^4	25	0.165	10^5	25	0.182
	100	0.025		100	0.051		100	0.078

Table 3.2: Posterior summaries of the time series simulation study. We consider different accuracies of the normalisation constant B and different proposal depth L. The table reports the Binder loss between the point estimate and the true latent partition of the data (BI). Results are averaged over 50 replicates.

We run each posterior sampling for 5 000 iterations of which 2 000 are discarded as burn-in period. To avoid label-switching problems, we obtain the point estimate of the latent partition $\hat{\lambda}$ as the one among those visited by the algorithm that minimizes the expected Binder loss function (Binder, 1978), following a decisional approach early introduced by Lau and Green (2007) and later studied by Wade and Ghahramani (2018). Table 3.2 shows the Binder loss function measured among the true and the estimated partition of the data, averaged over the 50 replicates, hence measuring how close are our estimates to the partitions that actually generated the synthetic data. Our algorithm performs consistently well with all configurations. Both the accuracy of the normalisation constants B and the proposal depth L seem to affect systematically its performance. Intuitively, large B and L should be preferred, as they increase the precision of the proposal and of its sampling step. However, small values of L strongly reduce the computational time needed to perform the sampling. Results also suggest that a large value of B is not needed, however if the number of realisations increases we suggest to increase also B.

We compare the performance of our model with competitor approaches, in terms of partition estimates. Regarding our proposal, we kept the previous specification of the model's parameters, with $B = 1\,000$ and $L = 25$, balancing computational time and accuracy. The first competitor we consider is an heuristic procedure, where we first estimate the change points marginally for each time series and then we group the observations resorting to a hierarchical clustering algorithm. The second competitor works in the spirit of k-means algorithms, looking at the dynamic time warping among different time series, here treated as functional observations. The third one is the *MixSeg* algorithm by Brault et al. (2024). Finally the *funFEM* algorithm by Bouveyron et al. (2015). We run each model on 50 replications of simulated data from the same study in Table 3.1. Similarly to the previous synthetic study, for each run and each model we evaluated the Binder loss function between the true and the estimated partitions. Small values mean that

the estimated partition is close to the true one. As shown in Figure 3.2, our model performs better than the others, producing more accurate estimates of the data clustering structure. Nevertheless, as for most competitors considered in the study, the accuracy of estimating the latent partition involves a moderate level of uncertainty, as evidenced by the dispersion observed in the boxplots of Figure 3.2.

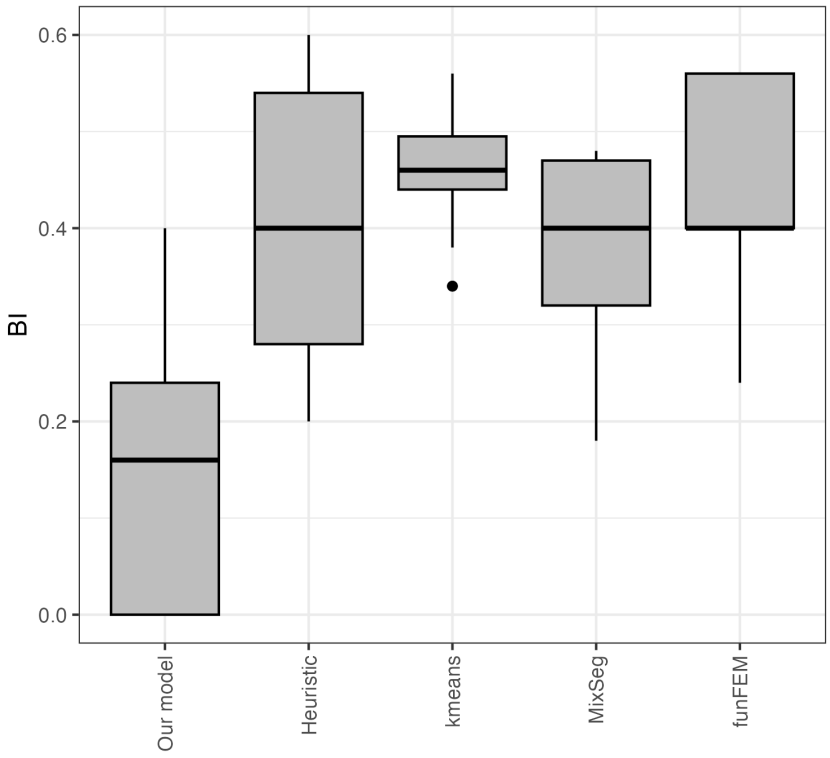


Figure 3.2: Binder loss (BI) distance between the estimated partition and the true partition over 50 replications of simulated data. Left to right: our proposal, an heuristic approach based on marginal estimates of change points, a distance-based approach among time series, the MixSeg model and the funFEM model.

3.5.1 Change point in currency rates

After presenting the properties of the model on synthetic time series, we provide an application to financial data. Specifically we consider the time series of the daily Euro exchange rate to the other main currencies[1]. An exchange rate is the value at which one currency can be exchanged for another one, its evolution is strongly determined by the economy of the countries that adopt that currency. We analyze the period that goes from 1st January 2022 to the 31st of January 2024, the total amount of time observations is 768 and the number of exchange rates is 29. We excluded the Russian Ruble, for which we are missing data for a long period of time, and the Bulgarian Lev. Time series have been marginally standardized before running the analysis. Further, we complete the model specification by setting $a = b = 1$, $c = 0.1$, and $\alpha = 1$. The

[1]Data are available in the European Central Bank website at https://www.ecb.europa.eu/stats/policy_and_exchange_rates/euro_reference_exchange_rates/html/index.en.html.

autocorrelation parameter is also fixed $\gamma = 0.1$, however we perform a sensitivity analysis without any evidence of changing in the results while varying γ, except for large values introducing a strong dependence over time. We set $B = 10\,000$, $L = 25$, and we sample $10\,000$, after a burn-in period of $5\,000$. Visual investigation of the sampled chains supports convergence of the algorithm.

We obtain a point estimate of the latent partition, as the one minimizing the expected Binder loss function, characterized by 9 clusters. Figure 3.3 shows the four largest clusters with their marginal change points. The largest cluster (bottom-right in Figure 3.3) includes the Polish Złoty, the Romanian Leu, the Singapore Dollar, the Thai Baht, the Turkish lira, the United States Dollar and the South African Rand. Euro exchange rate with these currencies remained quite stable during the analyzed period, no marginal change points are in fact detected. The second largest cluster (top-right in Figure 3.3) includes the Hong Kong Dollar, the Japanese Yen, the Mexican Peso, the Norwegian Krone, the New Zealand Dollar and the Philippine peso. They share a change point around June 2022, January 2023, June 2023 and April 2023. One of the other two clusters (top-left in Figure 3.3) contains the Swiss Franc, the Czech Coruna, the Hungarian Forint and the Israeli Shekel. The other one (bottom-left in Figure 3.3) includes the British pound sterling, the Icelandic Króna, the Malaysian Ringgit and the Swedish Krona. The marginal change points of these two groups occur at similar times, some of them are very close, others differ only for one or two months. Details about the remaining smaller clusters are shown in Appendix B.3.

3.6 Clustering epidemics with similar structural changes

Next, we apply the previous method to epidemic data. Hereby, we aim to model jointly epidemic spread in different areas, while clustering similar regions on the basis of structural changes in the spreading of the disease. To this end, we consider the standard compartmental susceptible-infected-removed (SIR) model, in which the population is segregated based on immunological statuses of the individuals. In an SIR model, individuals belong to exactly one of the three compartments of susceptible (S), infected and infectious (I) or recovered/removed (R) at any given time point. Under the stochastic law of mass-action (Andersson and Britton, 2000), an infected individual infects susceptible individuals whenever an individual Poisson clock rings, before eventually recovering. Once recovered, they play no role in the dynamics of the disease spread.

Let $\boldsymbol{X}(t) = (X_S(t), X_I(t), X_R(t))$ where the stochastic processes $X_S(t), X_I(t)$, and $X_R(t)$ denote the numbers at time t of susceptible, infected and recovered individuals respectively, and take values in the set of non-negative integers $\mathbb{Z}_+$. Since no birth or immigration is assumed into the population, the total population size is conserved at all times, i.e. $X_S(t) + X_I(t) + X_R(t) = X_S(0) + X_I(0) + X_R(0)$ at all times $t \geq 0$. As we are primarily interested in large populations, we use the quantity

$$\varepsilon = \frac{1}{X_S(0)} \in (0, \infty) \tag{3.12}$$

as a scaling parameter and study the behaviours of the system as $\varepsilon \to 0$. We further assume

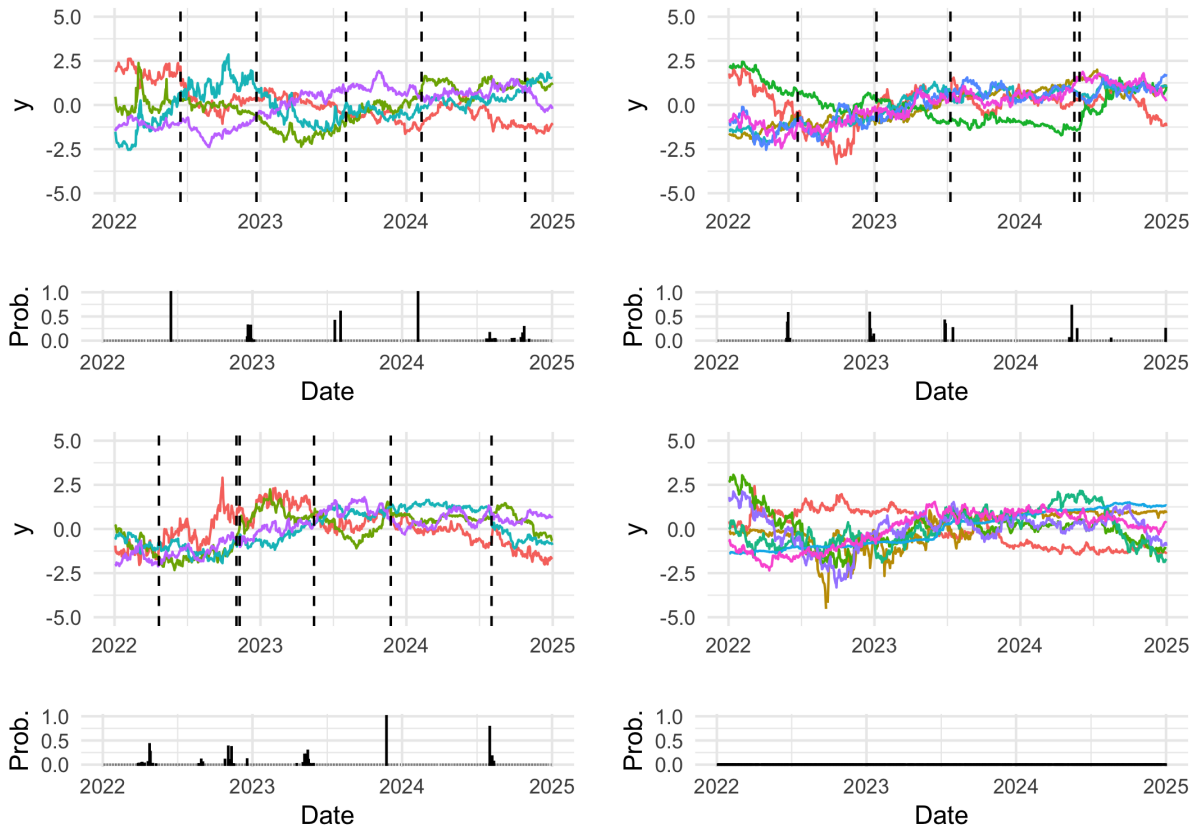


Figure 3.3: Four largest clusters in the Euro exchange rate data application. In each block, the bottom histogram represents the posterior probability of having a change point at each specific time, while the upper part shows the currencies exchange rates of each cluster, with the cluster-specific marginal change points denoted by the dashed lines.

$\varepsilon X_I(0) \to I_0 \in (0,1)$ as $\varepsilon \to 0$, where the limiting quantity I_0 is non-random (deterministic), and assume the initial number of recovered individuals $X_R(0)$ is zero. Let $\beta(t)$ and $\xi(t)$, bounded functions of time t, represent the time-varying infection and recovery rates respectively. Let us consider the scaled stochastic process $\boldsymbol{X}^{(\varepsilon)}(t) = (X_S^{(\varepsilon)}(t), X_I^{(\varepsilon)}(t), X_R^{(\varepsilon)}(t))$, where $X_S^{(\varepsilon)} = \varepsilon X_S, X_I^{(\varepsilon)} = \varepsilon X_I$, and $X_R^{(\varepsilon)} = \varepsilon X_R$. The scaled stochastic process $\boldsymbol{X}^{(\varepsilon)}(t)$ is a continuous-time Markov process satisfying

$$
\begin{aligned}
X_S^{(\varepsilon)}(t) &= X_S^{(\varepsilon)}(0) - M_S^{(\varepsilon)}(t) - \int_0^t \beta(u) X_S^{(\varepsilon)}(u_-) X_I^{(\varepsilon)}(u_-) \mathrm{d}u, \\
X_I^{(\varepsilon)}(t) &= X_I^{(\varepsilon)}(0) + M_I^{(\varepsilon)}(t) + \int_0^t \beta(u) X_S^{(\varepsilon)}(u_-) X_I^{(\varepsilon)}(u_-) \mathrm{d}u - \int_0^t \xi(u) X_I^{(\varepsilon)}(u_-) \mathrm{d}u, \\
X_R^{(\varepsilon)}(t) &= X_R^{(\varepsilon)}(0) + M_R^{(\varepsilon)}(t) + \int_0^t \xi(u) X_I^{(\varepsilon)}(u_-) \mathrm{d}u,
\end{aligned}
$$

where u_- denotes the left-hand limit at u and the stochastic processes $M_S^{(\varepsilon)}, M_I^{(\varepsilon)}$, and $M_R^{(\varepsilon)}$ are square-integrable zero-mean martingales, which converge to the zero function in probability as $\varepsilon \to 0$. See Section B.2 of the Appendix for a sketch of the derivation of these trajectory equations. Let $||(x_1, x_2, x_3)||_\infty = \max\{|x_1|, |x_2|, |x_3|\}$. As consequence of the functional law of large numbers, it can be proven that

$$
\lim_{\varepsilon \to 0} \sup_{t \le T} ||\boldsymbol{X}^{(\varepsilon)}(t) - \bar{\boldsymbol{X}}(t)||_\infty \xrightarrow{a.s.} 0 \tag{3.13}
$$

where $\bar{\boldsymbol{X}}(t) = (S(t), I(t), R(t))$ is the solution of the following system of ordinary differential equations

$$
\frac{\mathrm{d}}{\mathrm{d}t} S = -\beta(t) S I, \qquad \frac{\mathrm{d}}{\mathrm{d}t} I = \beta(t) S I - \xi(t) I, \qquad \frac{\mathrm{d}}{\mathrm{d}t} R = \xi(t) I, \tag{3.14}
$$

with the initial condition $S(0) = 1, I(0) = I_0$, and $R(0) = 0$, and some $T > 0$ (see, e.g., Theorem 3.1 of KhudaBukhsh and Rempała, 2024, for a proof).

We note from (3.13) that the cumulative infection pressure $H_\varepsilon(t)$ converges to a deterministic limit, satisfying

$$
H_\varepsilon(t) = \int_0^t \beta(u) X_I^{(\varepsilon)}(u) \mathrm{d}u \xrightarrow{a.s.} H(t) = \int_0^t \beta(u) I(u) \mathrm{d}u.
$$

Therefore, by virtue of the Sellke construction (e.g., see Andersson and Britton, 2000, and also Section B.2 of the Appendix), in the limit of the large population the probability that a randomly chosen susceptible individual is still susceptible at time t is given by $\exp(-H(t))$, which is precisely the function $S(t)$. That is, the function $S(t)$ can be interpreted as an improper survival function (improper since $\lim_{t\to\infty} S(t) > 0$) describing the random variable time to infection of a randomly chosen susceptible individual. This forms the basis of the so called dynamical survival analysis (DSA, KhudaBukhsh et al., 2023; Rempała and KhudaBukhsh, 2023). In practice, we can condition on a final observation time T to get a proper survival function, which admits a

density

$$f_T(t) = -\left(\frac{1}{1-S(T)}\right)\frac{\mathrm{d}}{\mathrm{d}t}S(t) = \frac{\beta(t)S(t)I(t)}{1-S(T)}. \tag{3.15}$$

The density in (3.15) describes the dynamics for a single population in continuous time. A discrete-time analog of the dynamical stochastic analysis method has indeed been developed in Wascher et al. (2024) in the context of monitoring the spread of a disease in a closed population such as a student cohort on campus. Since we are interested in applying our method to data at the level of countries, we decide in favour of a simple model. In practice, this means we will discretise time and use the time-discretised model for clustering purposes at equally-spaced times $t \in \{1, \dots, T\}$, which in the following corresponds to different days, whereas the infection rate parameter of the generic ith observed country $\boldsymbol{\beta}_i = \{\beta_{i,1}, \dots, \beta_{i,T}\}$ changes at certain times described by the specific latent order of the data, while the remaining parameters are assumed to be homogeneous over time. Therefore, given a random sample of daily cases of infections $y_{i,1}, \dots, y_{i,T}$ at a generic ith location, observed up to some terminal time $T > 0$, the likelihood contribution of the ith observed series is given by

$$\mathcal{L}(\boldsymbol{y}_i \mid \boldsymbol{\beta}_i, \rho_i) = \prod_{j=1}^{k_i} \prod_{t=t_{i,j}^-}^{t_{i,j}^+} f_{i,T}(t)^{y_{i,t}}, \quad i = 1, \dots, n, \tag{3.16}$$

where $f_{i,T}(t)$ is defined in (3.15) and obtained by integrating numerically the system in (3.14) with infection rate $\boldsymbol{\beta}_i$. This is the likelihood function we will use for our clustering inference.

We remark that, also for the epidemiological case, we are not interested in inferring parameter values but on the latent clustering structure of the data. However, while in Section 3.5 we could integrate analytically the parameters of the kernel function, here we resort to numerical integration of those parameters. Specifically, we assume the recovery rate ξ to be constant and shared across different infection time series. The parameter I_0 can be updated through an observation-specific Metropolis–Hastings step. Finally, the infection rates $\boldsymbol{\beta}_i$s, which change over time, are marginalised via Monte Carlo integration.

3.6.1 Synthetic infection study

We first show through a simulation study the performance of the proposed model in an epidemiological scenario. Data are generated using the Doob–Gillespie algorithm (see Anderson and Kurtz, 2015a, and Appendix B.3 for pseudo-code). We consider a set of $n = 10$ populations, where the starting number of susceptibles is $S_0 = 100\,000$. We keep the recovery rate fixed at $\xi = 1/8$, we consider a time-varying rate of infection $\beta_{i,t}$ and a different starting proportion of infected $I_{i,0}$ for each population. The simulated data are generated in a time period of 200 days, however in the analysis we consider a restricted window of time to remove the tails of the epidemic. We restrict the window from time $t = 10$ to time $t = 150$ thus resulting in a time period of length $T = 140$. We assume three different groups where the parameters are as shown in Table B.1. In the first group we simulate an epidemic where we have two large spreads, while in the second and third group we simulate a single large spread but with different change points.

MC = 250			MC = 500			MC = 1 000		
B	L	BI($\hat{\lambda}, \lambda_0$)	B	L	BI($\hat{\lambda}, \lambda_0$)	B	L	BI($\hat{\lambda}, \lambda_0$)
	1	0.345		1	0.200		1	0.005
1 000	25	0.294	1 000	25	0.144	1 000	25	0.010
	100	0.257		100	0.194		100	0.018
	1	0.354		1	0.234		1	$\ll 10^{-4}$
10 000	25	0.323	10 000	25	0.042	10 000	25	0.010
	100	0.281		100	0.162		100	0.008
	1	0.320		1	0.280		1	0.014
100 000	25	0.310	100 000	25	0.249	100 000	25	0.007
	100	0.212		100	0.092		100	0.009

Table 3.3: Posterior summaries of the synthetic epidemiological data. Different scenarios are obtained combining different accuracies of the numerical integration of the infection rates MC, different accuracies of the normalisation constant B and different proposal depths L. The table reports the Binder loss between the point estimate and the true latent partition of the data (BI). Results are averaged over 50 replicates.

Figure 3.4 shows an example of simulated data for each of the groups considered in the study.

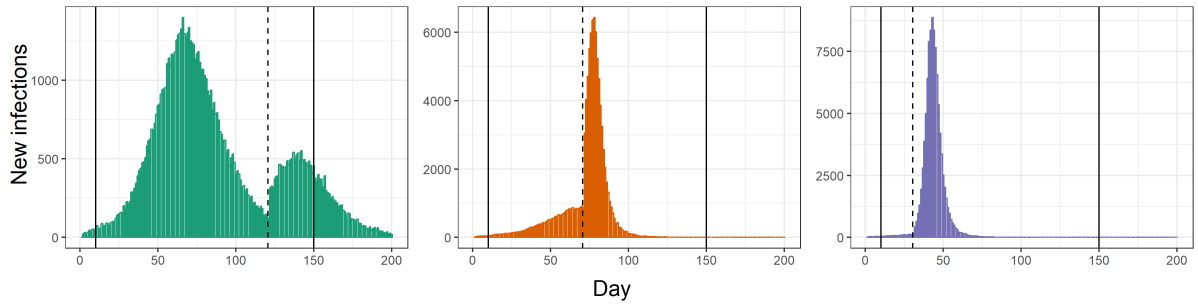


Figure 3.4: An example of bar plots of daily new cases in the synthetic study. Different panels correspond to different populations, one for each cluster considered in the study.

We investigate how different parameters affect posterior estimates and the algorithm performance. Specifically, we vary the number of points used to marginalise via Monte Carlo the observation-specific infection rates, here denoted by MC, as well as the accuracy of the normalisation constant B and depth of the proposal L. We set $\xi = 1/8$ to be consistent with the parameters we used to generate synthetic data. Further, we marginalise the infection rate whereas the Monte Carlo integration is performed by sampling the unique values in $\boldsymbol{\beta}_i$ from a gamma distribution with $(4, 10)$ as shape and rate parameters. Finally, for each population we randomly sample without replacement 20% of the infection times, since the dynamical survival analysis method requires only a subset of infections (see KhudaBukhsh et al., 2023). In each replicate, we run the algorithm for 5 000 iterations, discarding 2 000 as burn-in period. The point estimate of the latent partition $\hat{\lambda}$ is then obtained as minimizer of the expected Binder loss function (Binder, 1978) among those visited by the algorithm. To assess the algorithm performance and the quality of the posterior estimates, we use the Binder loss between the posterior point estimates and the true latent partitions of the data, reported in Table 3.3.

We can appreciate a similar behaviours to the one observed in the time series synthetic study: also with a complex kernel function, such as an SIR model, posterior estimates and performance of the algorithm are not particularly affected by the normalisation constant accuracy B and the proposal depth L. However, increasing the number of points used for the Monte Carlo marginalisation of the infection rate (MC) has a positive effect on the quality of the partition estimates. In particular, for $MC = 1000$ we obtain posterior estimates having comparable precision to the time series synthetic study (Section 3.5), where local parameters are marginalised analytically.

3.6.2 SARS-CoV-2 Europe infection data

We now consider the real data analysis motivating the model we study in this manuscript. Our aim is to identify possible clusters of homogeneous states among the 27 countries that comprise the European Union, with respect to the spread of SARS-CoV-2 virus. The first confirmed case was in Bordeaux (France) on the 24th January 2020, while the first major outbreak was experienced in Italy in late February, and different countries within the European Union experienced different starting time of the spreading. Although the response to the pandemic was different from one country to another, some actions were taken jointly with the European commission such as travel ban and vaccination.

Hereby, we consider infection data taken from the European Centre for Disease Prevention and Controls (2022) database on COVID-19 daily new cases, for a time period of more than one year that goes from 1st June 2021 to 30th September 2022. We selected this specific time window to mitigate differences of the pandemic, as it starts after the beginning of the world mass immunization campaigns and includes the largest spread of new detected case in Europe that lasted from the end of 2021 to September 2022. We smooth the data by computing for each day the 7-days rolling average of the number of infected. By doing so, we remove some common measurement errors that occur when counting the number of daily new cases, e.g. the number of weekend recorded daily new cases is always smaller than during week days. We set $\xi = 1/6$ and the unique values of the infection rates out of each $\boldsymbol{\beta}_i$ are sampled from a gamma distribution with $(2, 10)$ as shape and rate parameters. We increase the accuracy of the normalisation constants to $B = 100\,000$, since we have $T = 487$ days, we set $L = 1$ to reduce the computational time, and $MC = 1\,000$. Finally, we do not consider the entire data, but a subset of 50 000 infection times randomly sampled for each country. The MCMC algorithm is run for 12 500 iterations of which 2 500 have been discarded as burn-in period.

The posterior point estimate of the latent partition of the data is composed of five clusters, which are shown in Figure 3.5, along with the posterior similarity matrix. We have a large cluster with 14 countries, two clusters with 5 countries, one with two countries and one with just one country. The posterior similarity matrix in Figure 3.5 shows that the clusters are well separated, moreover, a visual check confirms convergence of the posterior chain. By analyzing the change point patterns of each cluster - available through the empirical survival functions in Appendix B.1 - we can infer the common features shared by countries grouped together. We identify four time periods in which most of the change points occur. The first is summer 2021 where the number of infections slightly increased after the curve was flattened by mass

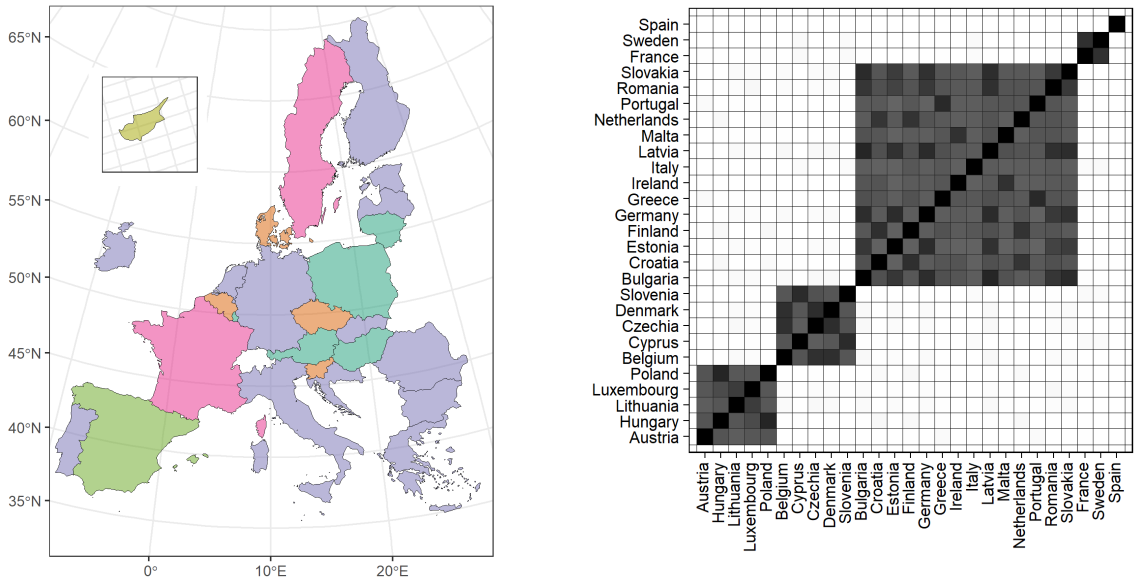


Figure 3.5: Summaries of the EU27 countries analysis. Left plot: countries colored consistently with the point estimate of the latent partition. Right plot: posterior similarity matrix.

vaccination, here all clusters exhibit at least one change point in this phase. The second period occurs in winter 2021-2022, when the Omicron variant started spreading. Clusters 1 and 2 have both a change point in January 2022, while Cluster 3 - the largest one - and Cluster 4 - which consists only of Spain - have an early change point in September when the number of infections gradually started to increase, reaching a peak at the beginning of 2022. In the same period France and Sweden experience two distinct change points, one in December and one in February. Later, in spring 2022, we can identify another set of change points; in particular, Clusters 3 and 4 have a change point in April and Cluster 3 in May. Finally in summer 2022 we detect change points for Spain, a highly popular tourist destination, as well as for Cluster 3 (which includes Italy and Greece) and Cluster 2.

3.7 Discussion

In the previous sections, motivated by an epidemiological study, we introduced a novel approach to cluster together time-dependent observations having the same structural change times. Such approach works on a mixture representation of the model, whereas the mixing parameters is a random order of the observational times, which induce locally homogeneous behaviours but also change points. The synthetic studies we presented have shown the performance of our proposal with respect to other competitors, favouring the first when performing model-based clustering on change points. Beyond the cases we considered in this chapter, this modeling approach can be easily adapted to several frameworks where the interest lies in defining groups of time-dependent observations.

Chapter 4

BayesChange: An R Package for Bayesian Change Point Analysis

4.1 Introduction

From the novel methodologies for change point analysis presented in the previous chapters, it clearly emerges that a key advantage of Bayesian methods in addressing this fundamental topic in modern statistics is their flexibility and rich modeling capabilities. In particular, they do not require the number or the locations of change points to be specified in advance and naturally allow for uncertainty quantification in the estimates. However, their main drawback lies in the theoretical and computational complexity, particularly in the implementation of MCMC sampling algorithms. Therefore, it is important to provide easy-to-use software that implements such methods.

In this chapter, we introduce the R package **BayesChange**, which implements Bayesian methods for change point analysis based on Product Partition Models (PPMs). Many models for change point analysis have been developed based on this framework, including those presented in Chapters 2 and 3, which represent some of the most recent advancements. **BayesChange** is written in C++ with an R (R Core Team, 2024) interface, providing Bayesian methods for change points analysis. The package offers three main contributions:

1. Implementation of the method by Martínez and Mena (2014) and the one introduced in Chapter 2 for change point detection in time series and survival functions associated with susceptibles-infected-removed (SIR) models.

2. Implementation of the method described in Chapter 3 for clustering time-dependent data with common change points.

3. S3 methods for graphical visualization and estimation of results from both change point detection and clustering.

While there are many R packages that perform change point detection, **BayesChange** implements procedures that are currently unavailable elsewhere. **BayesChange** not only provides implementations of methods that are not available in other packages, but also introduces novel and more flexible approaches for change point detection. The method by Martínez and Mena (2014)

and the one introduced in Chapter 2, thanks to the Bayesian framework, allow change point detection in univariate and multivariate time series without requiring the user to pre-specify the number of change points. Moreover, they naturally provide uncertainty quantification for both the number and locations of the change points through the posterior distribution. The model proposed in Chapter 3, on the other hand, addresses a problem that no other models currently solve: clustering time-dependent data based solely on shared change point locations as the unique criterion for commonality. Furthermore, methods in **BayesChange** are all based on the PPM theory. According to this framework, realisations are associated with a sequence of latent parameters, possibly with ties. If the parameter changes value, then a change point occurs. Consecutive realisations sharing the same parameter form a block, resulting in a latent order induced by the change points. The probability of this latent order is proportional to the product of functional of the block sizes. The **bcp** package by Erdman and Emerson (2007) implements the change point detection procedure by Barry and Hartigan (1993), but unfortunately it has been recently archived from the Comprehensive R Archive Network. The function `cpp_ppm` of **ppmSuite** (Page et al., 2023) performs change point detection for univariate and multivariate time series, following the methodology of Quinlan et al. (2024), assuming a correlation structure between change points. Finally, package **HDcpDetect** (Okamoto et al., 2018) is not based on product partition models, but implements two Bayesian methods for change point detection. Specifically, a bisection algorithm strategy and an approach based on testing a large number of possible change point structure, selecting the one that best fits the data.

Even though our main interest is on the Bayesian framework, we mention some packages that perform change point detection with a frequentist approach. The **decp** package (Pavlopoulos et al., 2024), recently made available on CRAN, provides two methods to infer changes on the covariance structure of multivariate time series, the **ecp** library (James and Matteson, 2014) includes two approaches for nonparametric change point detection on multivariate data and the **cpm** package (Ross, 2015) performs both detection and prediction of change points on univariate time series. There are also packages that do not perform detection of change points, but take into account changes in the data when performing other analysis, like the **mixtools** package by Benaglia et al. (2009), that performs regression modeling possibly accommodating for change points in the predictors. Finally, since **BayesChange** includes also a method for clustering survival functions arising from SIR models that share the same changes. We also mention **CPsurv** (Krügel et al., 2017) which detects change points on survival data using a nonparametric approach.

The chapter is structured as follows. Section 4.2 introduces the methodologies implemented in **BayesChange**. Section 4.3 provides details on the functions and the S3 methods of the package. Finally, Section 4.4 shows how to use **BayesChange** in practice, trough synthetic data examples. Additional materials, such as details on the algorithms and the data generating processes, can be found in Appendix C.

4.2 Models and posterior estimates

In this section, we introduce the theory beyond the methods included in the package. **BayesChange** includes a total of six different methodologies, to carry out different types of change point analysis. Three of these perform change point detection respectively on univariate time series, multivariate time series and on a single survival function arising from an SIR model. The other three methods perform model-based clustering of samples of the previous, on the base of common change points.

Following the previous chapters, a single observation is denoted as a vector $\boldsymbol{y}_i = (y_{i,1}, \dots, y_{i,T})$ consisting of T realisations. If the observation is a time series, then $y_{i,t} \in \mathbb{R}^d$, with $d = 1$ for the univariate case and $d > 1$ for the multivariate one. In the context of an epidemic diffusion derived from an SIR model, $y_{i,t} \in \mathbb{N}^+$, as $y_{i,t}$ represents the number of newly infected individuals in population i on day t. The data-generating distribution $f(\cdot \mid \theta_{i,t})$ depends on the time-dependent parameter $\theta_{i,t}$, which characterizes the local behaviour of $\boldsymbol{y}_i$. A change point in the time series occurs whenever $\theta_{i,t}$ changes. The generic jth block of the ith latent order is associated with an unique value of the latent parameter $\boldsymbol{\theta}^*_{i,j}$. Two observations $y_{i,t}$ and $y_{i,\ell}$ belong to the generic jth block if they share the same value of such parameter, $\theta_{i,t} = \theta_{i,\ell} = \theta^*_{i,j}$. Detecting change points in $\boldsymbol{y}_i$ means finding a point estimate for ρ_i, while clustering time dependent data with common change points means grouping together observations with the same latent orders. In the following subsections we present in details the two procedures.

4.2.1 Detecting change points on time series

In this package, methods for detecting change points on time series are based on the model by Martínez and Mena (2014) and the one described in Chapter 2. The former proposes a Bayesian nonparametric procedure to detect change points while the latter is a generalization of the first method to multivariate time series. In both cases, following Martínez and Mena (2014), the prior probability distribution of the latent order ρ_i is devised by restricting the exchangeable partition probability function (EPPF) of a Pitman-Yor process to the space of orders. Hence, the resulting distribution of the latent order equals

$$\mathcal{L}(\rho_i) = \frac{T!}{k_i!} \frac{\prod_{j=1}^{k_i-1}(\delta + j\sigma)}{(\delta+1)_{T-1}} \prod_{j=1}^{k_i} \frac{(1-\sigma)_{m_{i,j}-1}}{m_{i,j}},$$

where $m_{i,j}$ denotes the cardinality of the jth block of $\boldsymbol{y}_i$, i.e. the number of observations assigned to such block, and $\sigma \in (0,1)$ and $\delta \in (-\sigma, \infty)$ denotes the discount and strength parameters of the Pitman-Yor process, respectively. With a straightforward application of the Bayes' rule, the posterior probability of ρ_i is then proportional to the product of the prior distribution $\mathcal{L}(\rho_i)$ and the likelihood of ρ_i given $\boldsymbol{y_i}$.

We assume a Markovian regime structure for $\boldsymbol{y}_i$. Specifically, if $d = 1$ the distribution of $y_{i,t}$ is given by an univariate Ornstein-Uhlenbeck process,

$$y_{i,t} \mid \mu, \lambda, \phi \sim OU(\mu, \lambda, \phi),$$

where $\mu \in \mathbb{R}$, $\lambda > 0$ and $0 < \phi < 1$. A priori, we assume a Normal-Gamma distribution for (μ, λ), with $\mu \mid \lambda \sim \mathrm{N}(0, (c\lambda)^{-1})$ and $\lambda \sim \mathrm{Ga}(a, b)$ for λ. Since we are not interested in the behaviour of the time series, but only in the position of change points, μ and λ are then integrated out of the likelihood, while ϕ is kept explicitly in the model. An explicit expression for the marginal likelihood is given by Martínez and Mena (2014) and reported in Section 2.2.1. If $d > 1$, we set as distribution of $y_{i,t}$ a multivariate Ornstein-Uhlenbeck process,

$$y_{i,t} \mid \boldsymbol{\mu}, \Lambda, \phi \sim OU(\boldsymbol{\mu}, \Lambda, \phi),$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$, Λ is a $d \times d$ matrix with nonnegative entries and $0 < \phi < 1$. Here, $(\boldsymbol{\mu}, \Lambda)$ are jointly distributed a priori as a Normal-Inverse Wishart distribution, with $\boldsymbol{\mu} \mid \Lambda \sim \mathrm{N}(\boldsymbol{m}_0, k_0\lambda)$ and $\Lambda \sim IW(\nu_0, S_0)$, where $\boldsymbol{m}_0 \in \mathbb{R}^d$, $k_0 > 0$, $\nu_0 > d-1$ and S_0 is a positive definite $d \times d$ matrix. Similarly to the univariate case, the parameters $\boldsymbol{\mu}$ and Λ are integrated out of the likelihood. Explicit calculations can be found in Section 2.2.1. In both univariate and multivariate cases, the law of the generic i-th observation is given by the product of the law of each block:

$$\mathcal{L}(\boldsymbol{y}_i \mid \rho_i, \boldsymbol{\theta}_i^*) = \prod_{j=1}^{k_i} \prod_{t_{i,j}^-}^{t_{i,j}^+} \mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta_{i,j}^*),$$

where $t_{i,j}^- = \min\{t : t \in A_{i,j}\}$ and $t_{i,j}^+ = \max\{t : t \in A_{i,j}\}$ are, respectively, the first and last time index of block $A_{i,j}$.

Posterior simulation

To obtain a posterior estimate of the latent order ρ_i, both methods implement the same Markov Chain Monte Carlo (MCMC) algorithm based on a split and merge scheme, in the spirit of Martínez and Mena (2014). The algorithm is inspired by the model introduced in Chapter 2 and detailed in Algorithm 3. At each step of this procedure with probability q a split is performed: a block is randomly chosen along with one observation that it belongs to. The block is then divided into two new blocks, with the chosen observation being the last realisation of the first block and the following observation the first of the new block. With probability $1 - q$ instead, a block is randomly chosen and its observations are merged with the observations of the following block. In both cases a new latent order is proposed where two blocks are merged together or an additional block is created. Once a new value for the latent order is proposed, the algorithm performs a Metropolis-Hastings step to accept or reject such a value. Finally, if the number of blocks in the proposed order is greater than 1, a shuffle step is performed in which observations of two adjacent blocks are rearranged while keeping the number of blocks unchanged. The output of the algorithm is a sequence of latent orders, one for each iteration, which represents a sample from the posterior distribution of ρ_i given a sequence of real-valued time-dependent observations. In addition, the algorithm includes posterior sampling for the restricted eppf parameters, σ and δ, and for the parameter ϕ. For parameters σ and ϕ at each step a new value is proposed and evaluated with a Metropolis-Hastings procedure, while for δ at each step a value is extracted from its full conditional probability, whose form is known.

The point estimate for the latent order is obtained by selecting from the posterior sample the order that minimizes a specific loss function, such as the Binder loss function (Binder, 1978) or the variation of information (Meilă, 2007; Wade and Ghahramani, 2018). For this purpose **BayesChange** also implements a method that selects the final order using a search algorithm called SALSO, introduced by Dahl et al. (2022).

Both methods for detecting change points were designed specifically for time series. However, **BayesChange** also provides a function to detect the change points on survival functions arising from an SIR model. Here, the marginal distribution of the data term is given by evaluating the density function associated with such a survival function, where local parameters appearing in the likelihood are integrated numerically. More details are provided in Section 4.2.3.

4.2.2 Clustering time-dependent data with common change points

BayesChange also includes methods for clustering time series or survival functions that share common change points. These methods are based on the novel methodology introduced in Chapter 3, which clusters time-dependent data that share the same change point locations, without assuming any other commonalities. As with the change point detection model, the underlying algorithm is the same for both time series and survival functions, differing only in the distribution and the likelihood of the data. We first start by presenting the method in the context of time series, and later we discuss its extension to epidemiological data.

Consider a set of time series $\mathcal{Y} = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_n\}$, where each realisation of the time series is distributed according to a distribution $f(\cdot \mid \theta_{i,t})$. The object of interest in this case is a random partition of $\{1, \dots, n\}$ in k groups, here denoted with $\lambda = \{B_1, \dots, B_k\}$, for which (i) $B_i \cap B_j = \emptyset$, for $i \neq j$, and (ii) $B_1 \cup \cdots \cup B_k = \{1, \dots, n\}$. Two generic observations $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ belong to the same group B_l if they have the same change points, so if their latent orders ρ_i and ρ_j correspond. The sequence $\mathcal{R}^* = \{\rho^\dagger_{(1)}, \dots, \rho^\dagger_{(k)}\}$ contains the unique latent orders associated with each block in λ. Thus, if $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ belong to the same block B_l, then $\rho_i = \rho_j = \rho^\dagger_{(l)}$. It is important to stress that the only commonality considered in this clustering method is the position of the change points, while the behaviour of the time series is not taken into account as clustering criterion. Time series generated by distributions with different parameters but where changes happen at the same times belong to the same group. The model implemented in this method is the one in Equation 3.3, where the distribution of $\boldsymbol{y}_i$ depends on its random order ρ_i and the block-specific parameters θ^*_i. The latent order ρ_i is sampled from a finite discrete mixture, where the weights are distributed according to a Dirichlet and the atoms are all the possible 2^{T-1} orders of $\boldsymbol{y}_i$. The parameters of the Dirichlet distribution are assumed to be all equal $\alpha_1 = \cdots = \alpha_{2^{T-1}} = \alpha$, which leads to a symmetric prior. The distribution $\mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta^*_{i,j})$ depends on the nature of the data we consider, if we consider univariate and multivariate time series, we use the same approach based on the Ornstein-Uhlenbeck process presented in Section 4.2.1. When we model epidemiological diffusion data, the distribution $\mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta^*_{i,j})$ is the likelihood of a survival function of an SIR model.

Posterior simulation

To obtain a posterior estimate of λ we implement the algorithm presented in Chapter 3. The procedure is still based on a split and merge procedure, but with the addition of a second level of sampling. Here, at each iteration both the latent partition of the observations λ and the unique latent orders in $\mathcal{R}^*$ are updated. At each step two observations are randomly chosen, if they belong to the same group a random split of this group is proposed. Alternatively, if they belong to different groups the two groups are merged in a single group. Then, we perform a Metropolis-Hastings step to accept the proposed partition. In order to evaluate the acceptance rate, we need to assign latent orders to the new blocks of the proposed latent partition. These latent orders, here denoted by ρ, are obtained using the instrumental proposal distribution (3.8), that is a mixture of the posterior distributions of the latent order conditionally on all observations. A proposal of this form assigns more mass to orders which are representative at least for a single observation, while having a flat proposal over the orders' space results in proposing rarely candidates which are suitable latent orders of the data. See Section 3.4 for further details. Finally, at the end of each iteration, a step called *acceleration step* updates all the latent orders in $\mathcal{R}^*$. Each element in $\mathcal{R}^*$ is updated with a procedure similar to the one introduced in Section 4.2.1, but conditionally on all observations to which the latent order is assigned. Algorithm 4 reported in Appendix C.1 provides the pseudocode of the clustering procedure for time series with common change points.

4.2.3 Extension to epidemiological data

Methods for change point detection and clustering of time dependent data with common change points in **BayesChange** are originally designed for real valued time series. However, they can be easily extended to other types of data by defining a proper likelihood function. For example, in Chapter 3 we consider the case where a generic observation $\boldsymbol{y}_i$ is a sequence of daily new infected individuals for the i-th population. The posterior inference procedure is similar to the one used for time series in both the change point detection and clustering methods. The only difference is that the likelihood must be defined according to the new kind of data. The likelihood is derived from the discretisations of a standard compartmental SIR model. The dynamic over time is described by the following differential equation system,

$$\frac{d}{dt}S = -\beta(t)SI, \qquad \frac{d}{dt}I = \beta(t)SI - \xi(t)I, \qquad \frac{d}{dt}R = \xi(t)I,$$

with the initial condition $S(0) = 0$, $I(0) = I_0$ and $R(0) = 0$. Here, $\beta(t)$ is the infection rate and $S(t)$ and $I(t)$ are respectively the number of susceptibles and infected individuals at time t. Hence, conditioning on a the final observational time, the density function associated to the survival function of the previous model has the form of Equation 3.15. This model is indexed by three time-dependent parameters: the vector of infection rates for a generic observation i, $\boldsymbol{\beta}_i = \{\beta_{i1}, \dots, \beta_{iT}\}$, that is time dependent and observation-specific, the starting proportion of infected individuals I_0 that is only observation-specific and the recovery rate ξ that is assumed constant over time. Since $f_T(t)$ is an intractable likelihood, these parameters cannot be integrated out analytically from the likelihood like in the time series application. As outlined in

Chapter 3, $\boldsymbol{\beta}_i$ is integrated with a Monte-Carlo procedure with the unique values sampled from a gamma distribution, I_0 is updated with a Metropolis-Hastings step and ξ is fixed.

4.3 Package structure

BayesChange provides two main R functions that call the C++ methods, with the techniques described in Section 4.2. These two functions are `detect_cp` and `clust_cp` that detect change points on time series or survival functions and cluster time series or survival functions with common change points, respectively. The previous functions are designed to be intuitive to use and require only a few mandatory arguments, without the need for an in-depth understanding of the underlying models. At the same time, users with a deep knowledge of the models can specify all the parameters of the models, tailoring function calls on specific scientific interests. To this aim, the functions have been written in an object-oriented framework with two S3 classes and methods that summarize and illustrate the final results.

We present in Section 4.3.1 the C++ implementation of the models, in Section 4.3.2 the R user interface and finally in Section 4.3.3 the general methods that can be applied to both S3 classes.

4.3.1 Implementation details

All functions performing posterior sampling contained in the **BayesChange** package are written in C++, mainly resorting to **Rcpp**, **RcppArmadillo** and **RcppGSL** libraries (Eddelbuettel and François, 2011; Eddelbuettel and Sanderson, 2014; Eddelbuettel and Francois, 2023). The latter is used mainly to sample efficiently from distributions, while the others are used to have computationally efficient samplers.

The detection of change points is performed by three functions, `detect_cp_uni` for univariate time series, `detect_cp_multi` for multivariate time series and `detect_cp_epi` for epidemiological diffusions. Both `detect_cp_uni` and `detect_cp_multi` consist of a `for` loop that is repeated for an arbitrary number of iterations. Within each iteration, the algorithm first randomly performs a split or merge of the latent order of the data, and then, if the number of blocks is larger than one, it also performs a shuffle step. To facilitate possible future extensions of the package, most fundamental operations within a single MCMC loop are coded in independent C++ functions. The functions `AlphaSplit_UniTS` and `AlphaSplit_MultiTS` compute the acceptance ratio of a split proposal, for univariate time series and multivariate time series, respectively. Similarly, `AlphaMerge_UniTS` and `AlphaMerge_MultiTS` compute the acceptance ratio for the proposal of a merge step, while `AlphaShuffle_UniTS` and `AlphaShuffle_MultiTS` for the proposal of a shuffle step. These functions return a value between 0 and 1. If the returned value is larger than a uniformly distributed random number taking values in $(0, 1)$, then the proposal is accepted as a new state of latent order. Otherwise, the previous configuration of the latent order is kept as the current state. Then, at the end of each iteration, functions `UpdateGamma`, `UpdatePhi` and `UpdateDelta` update the main parameters of the model. The function `detect_cp_epi` has been designed in a slightly different way to optimize computational time, since computing the likelihood of survival functions arising from SIR models is more de-

manding. The function is composed of a `for` loop. At each iteration, the function `update_I0` updates the proportion of infected individuals at time zero, assumed to be unknown, and the function `update_single_order` updates the change points of the epidemiological diffusion.

The **BayesChange** package performs clustering of time-dependent data with common change points with three main functions. Similarly to change point detection, `clust_cp_uni` handles univariate time series, `clust_cp_multi` is built for multivariate time series and `clust_cp_epi` for epidemics. All these methods start with a function that computes an approximation of the normalisation constant for the mixture in Equation 3.8 with the given data. Specifically, `norm_constant_uni`, `norm_constant_multi` and `norm_constant_epi` compute the normalisation constant for univariate time series, multivariate time series and SIR survival functions, respectively. Then, the main `for` loop run for an arbitrary number of iterations. For functions `clust_cp_uni` and `clust_cp_multi`, the proposals at each step are evaluated through functions `AlphaSplit_Clust` and `AlphaMerge_Clust`, analogously to the change point detection case. Before evaluating the acceptance ratio, a new latent order is assigned to each new proposed group. This is done with `SplitMergeUniTS` and `SplitMergeMultiTS`. These functions have the same structure of `detect_cp_uni` and `detect_cp_multi`, but of void type, to save memory usage. At the end of each MCMC iteration, an acceleration `SplitMergeUniTS` performs an acceleration step for univariate time series and `SplitMergeMultiTS` for multivariate time series. The function `clust_cp_epi` has been designed differently, for the same reason as `detect_cp_epi`. The MCMC loop includes `update_I0`, `update_partition`, which updates the partition of the data, and `update_single_order`, where the latter is applied to all the unique values of the latent order to perform the acceleration step.

4.3.2 User interface

We provide here a description of the R user interface of **BayesChange**. Two wrappers, `detect_cp` and `clust_cp`, have been designed to interact with the C++ functions presented in Section 4.3.1.

Detect change points

With function `detect_cp` the user can perform change point detection on univariate or multivariate time series and on epidemic diffusions. The first argument of this function is `data`. It can be either a vector, if we analyze an univariate time series or an epidemic diffusion, or a matrix, when we deal with a multivariate time series. The user specifies with the `kernel` argument the type of data considered in the analysis. If `kernel = "ts"` the algorithm automatically detects if `data` is a vector or a matrix and calls the C++ functions `detect_cp_uni` or `detect_cp_multi`. If `kernel = "epi"` then the algorithm calls `detect_cp_epi`. A representation of this process is given by the diagram in Figure 4.1.

Each call of the `detect_cp` function can be tuned trough the following arguments:

- `n_iterations`: number of MCMC iterations.
- `n_burnin`: the number of burn-in iterations. By default `n_burnin = 0`.
- `q`: the probability of performing a split at each step. By default `q = 0.5`.

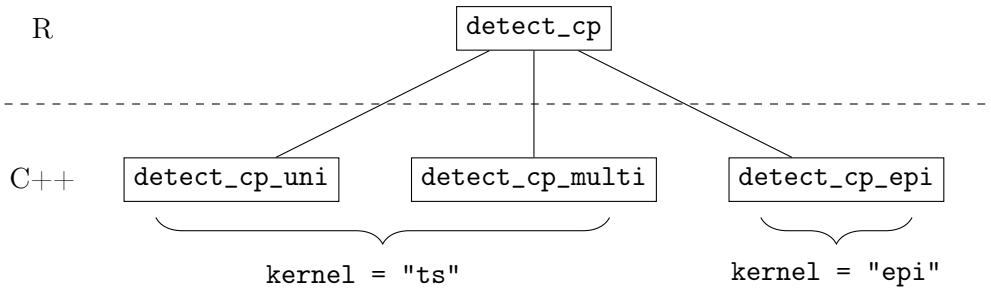


Figure 4.1: Diagram representation of function `detect_cp` call for different type of data.

- `params`: a list with the parameters specific for the chosen kernel. On Table 4.1 are detailed the arguments for both time series (univariate or multivariate setting) and epidemiological diffusions.
- `kernel`: set `kernel = "ts"` if data are time series, set `kernel = "epi"` if data are infections from an epidemic.
- `print_progress`: if `TRUE` print the progress of the algorithm.
- `user_seed`: the seed for the random distribution generation.

Function `detect_cp` returns an object of class `"DetectCpObj"` that contains the following objects:

- `data`, `n_iterations` and `n_burnin`: these objects contain respectively the data, the number of iterations and of burn-in steps specified by the user.
- `orders`: a matrix in which are stored the latent orders sampled at each iteration. Each row corresponds to an iteration and each column to a realisation of the latent order.
- `time`: computational time of the algorithm in seconds.
- `phi_MCMC` and `phi_MCMC_01`: if data are time series; these two objects respectively store the vector containing the posterior samples from the Metropolis-Hastings updates of ϕ, and a binary vector indicating whether the proposed value of ϕ was accepted (1) or rejected (0) at each iteration.
- `sigma_MCMC` and `sigma_MCMC_01`: if the data are time series; these two objects respectively store the vector containing the posterior samples from the Metropolis-Hastings updates of ϕ, and a binary vector indicating whether the proposed value of σ was accepted (1) or rejected (0) at each iteration.
- `delta_MCMC`: if data are time series; a vector with posterior sample of δ.
- `I0_MCMC` and `I0_MCMC_01`: if data are time series; these two objects respectively store the vector containing the posterior samples from the Metropolis-Hastings updates of I_0, and a binary vector indicating whether the proposed value of I_0 was accepted (1) or rejected (0) at each iteration.

- `kernel_ts` and `kernel_epi`: boolean objects equal to `TRUE` if data are respectively time series or epidemic diffusions.
- `univariate_ts`: if data are time series; a boolean object equal to `TRUE` if data is an univariate time series, `FALSE` if it is a multivariate time series.

Model	Argument	Interpretation
Univariate Time Series	`a`	parameters of the prior $\mathrm{N}(0,(c\lambda)^{-1})\mathrm{Ga}(a,b)$ for μ and λ
	`b`	
	`c`	
	`prior_var_phi`	variance σ^2_ϕ in the proposal $N(0,\sigma^2_\phi)$ for the estimate of ϕ
	`prior_delta_c`	parameters of the full conditional distribution of δ
	`prior_delta_d`	
Multivariate Time Series	`m_0`	parameters of the prior $NIW(\boldsymbol{m}_0,k_0,\nu_0,S_0)$ for $\boldsymbol{\mu}$ and Λ
	`k_0`	
	`nu_0`	
	`S_0`	
	`prior_var_phi`	variance σ^2_ϕ in the proposal $N(0,\sigma^2_\phi)$ for the estimate of ϕ
	`prior_delta_c`	parameters of the full conditional distribution of δ
	`prior_delta_d`	
Epidemic Diffusions	`M`	number of Monte Carlo iterations for the likelihood integration
	`xi`	recovery rate ξ
	`a0`	parameters of the gamma proposal used to integrate out the infection rates
	`b0`	
	`I0_var`	variance in the $N(0,\sigma^2_{I_0})$ proposal for updating I_0

Table 4.1: Parameters for the list of arguments in `params` of `detect_cp`

Clustering data with common change points

`clust_cp` includes the three methods to perform clustering of time-dependent observations with common change points. The structure is similar to that of `detect_cp`. The function still includes arguments `data`, `n_iterations`, `n_burnin`, `q`, `kernel`, `print_progress` and `user_seed`, with slight differences for `data` and `q`. For univariate time series or epidemic diffusions, the `data` argument is a matrix where each row is an observation and each column a realisation. If data are multivariate time series, `data` is a multidimensional array, where each slice is a matrix with rows denoting the dimensions and columns the realisations. The parameter `q` also denotes the probability of a split, but here for the acceleration step and in the split and merge proposal step of the algorithm. The argument `params` contains parameters specific for each type of data, detailed in Table 4.2. Based on the specified `kernel` and the data type, `clust_cp` calls `clust_cp_uni`,

Model	Argument	Interpretation
Univariate Time Series	`a`	parameters of the prior $\mathrm{N}(0,(c\lambda)^{-1})\mathrm{Ga}(a,b)$ for μ and λ
	`b`	
	`c`	
	`phi`	correlation parameter in $OU(\mu,\lambda,\phi)$.
Multivariate Time Series	`k_0`	parameters of the prior $NIW(\boldsymbol{m}_0,k_0,\nu_0,S_0)$ for $\boldsymbol{\mu}$ and Λ
	`nu_0`	
	`S_0`	
	`m_0`	
	`phi`	correlation parameter in $OU(\boldsymbol{\mu},\lambda,\phi)$
Epidemic Diffusions	`M`	number of Monte Carlo iterations for the likelihood integration
	`xi`	recovery rate ξ
	`a0`	parameters of the gamma proposal used to integrate out the infection rates
	`b0`	
	`I0_var`	variance in the $N(0,\sigma^2_{I_0})$ proposal for updating I_0
	`avg_blk`	average number of blocks when random orders are generated

Table 4.2: Parameters for the list of arguments `params` of `clust_cp`

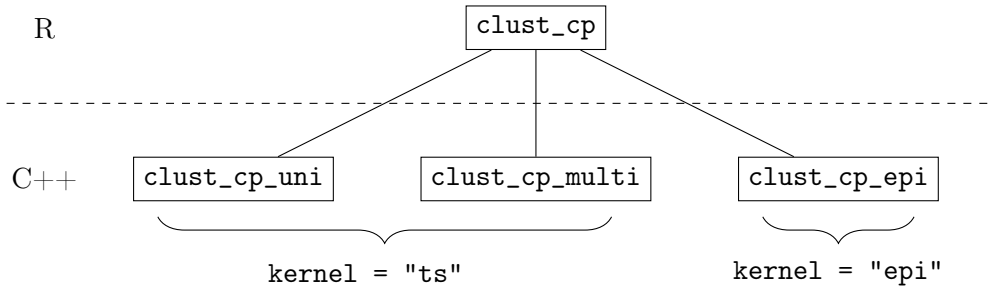


Figure 4.2: Diagram representation of function `clust_cp` call for different type of data.

`clust_cp_multi` or `clust_cp_epi`, according to the diagram in Figure 4.2. Further, the new arguments in `clust_cp` are the following:

- `alpha_SM`: the parameter value of the symmetric Dirichlet distribution in Equation 3.3.
- `B`: the number of latent orders randomly generated when the approximation of the normalisation constants is computed.
- `L`: the number of split and merge steps performed to propose a latent order for the new groups.

The output of `clust_cp` is an object of class `"ClustCPObj"`. Besides the inputs of the user, saved in `data`, `n_iterations` and `n_burnin` , it contains the following elements:

- `clust`: a matrix where each row corresponds to the latent partition of the data, returned at each iteration of the MCMC algorithm.

- `orders`: an array where each element is a matrix with the unique latent orders of the proposed partition, specific for each iteration.
- `time`: computational time of the algorithm in seconds.
- `norm_vec`: the vector with the observation-specific contribution of the approximated normalisation constant.
- `I0_MCMC` and `I0_MCMC_01`: if the data are epidemic diffusions, these two objects respectively store the vector containing the posterior samples from the Metropolis-Hastings updates of I_0, and a binary vector indicating whether the proposed value of I_0 was accepted (1) or rejected (0) at each iteration.
- `kernel_ts` and `kernel_epi`: boolean objects equal to `TRUE` if data are respectively time series or epidemic diffusions.
- `univariate_ts`: only if `kernel_ts = TRUE`, if `TRUE` time series are univariate, `FALSE` if they are multivariate.

4.3.3 Generic methods and additional functions

BayesChange provides S3 methods for both `"DetectCpObj"` and `"ClustCpObj"` objects. These methods summarize the output of functions `detect_cp` and `clust_cp`, give information about the algorithm, provide a point estimation method and a graphical illustration of the results.

The first method is `print`, it returns a message that says which kind of data have been analyzed and what type of algorithm has been run. It says if the algorithm is for change point detection or clustering of time dependent data, and also if data are univariate time series, multivariate time series or epidemic diffusions. Method `summary` returns the same information of `print` and in addition the number of iterations, of burn-in steps and the computational time in seconds. Method `posterior_estimate` provides a point estimate by making use of package **salso**. This package is based on the search algorithm `salso` by Dahl et al. (2022) which provides a point estimate for a random partition. We included this dependence because the **salso** package implements several loss functions and embeds other popular methods for estimating latent partitions as special cases. The first argument of `posterior_estimate` is an object of class `"DetectCpObj"` or `"ClustCpObj"`, the other arguments are the same of function `salso`. When performing change point detection, i.e. for `"DetectCpObj"` objects, the function returns an estimate of their locations. When clustering time-dependent quantites, i.e. for `"ClustCpObj"` objects, the function returns an estimate of the latent partition of the data. Finally, the `plot` method has been extended resorting to the **ggplot2** package. It takes as arguments the same of `posterior_estimate`, since it first computes a point estimate before providing a graphical representation.

BayesChange includes also the function `sim_epi_data`. Such a function generates synthetic survival data using the Doob–Gillespie algorithm. See Anderson and Kurtz (2015b) for further details. The output of the function is a vector with the simulated infection times. The arguments of `sim_epi_data` are the following:

- `S0`: number of individuals in the population.
- `I0`: number of infected individuals at time zero.
- `max_time`: maximum observed time.
- `beta_vec`: vector of time-dependent infection rates.
- `xi_0`: recovery rate.
- `user_seed`: the seed for the random distribution generation.

4.4 Illustrations

In this section, we guide the reader through the use of the main methodologies provided by the **BayesChange** package. We show how to detect change points in time series and epidemic diffusions with `detect_cp`, and how to cluster time-dependent data that share the same change points with `clust_cp`. To illustrate all functions in a reasonable computational time, we decide here to implement **BayesChange** on synthetic data. For each function, we provide the code to generate synthetic data, which helps to illustrate which kind of input object is required to run each specific algorithm. In the context of epidemic diffusions, we also show how to simulate data with `sim_epi_data`. For each illustration, we provide the code and the plot of the final estimate.

4.4.1 Change point detection on time series and survival functions

At first, we show how to run an analysis with the `detect_cp` function on all different types of data **BayesChange** can handle. Since models implemented in **BayesChange** are capable of detecting change both in the mean and in the variance, we generate synthetic data with a Gaussian distribution by changing both the local mean and variance values when a change point occurs. We simulate a time series with 200 realisations and two change points, respectively located at times 51 and 151.

```
R> data <- as.numeric()
R> data[1] <- rnorm(1, mean = 0, sd = 0.13)
R> for(i in 2:50){
R>  data[i] <- 0.1 * data[i-1] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.13)}
R> data[51] <- rnorm(1, mean = 1.5, sd = 0.15)
R> for(i in 52:150){
R>  data[i] <- 0.1 * data[i-1] + (1 - 0.1) * 1.5 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.15)}
R> data[151] <- rnorm(1, mean = 0, sd = 0.12)
R> for(i in 152:200){
R>  data[i] <- 0.1 * data[i-1] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.12)
R> }
```

Before running the algorithm, we need to set the specific parameter for the univariate model by defining a list with the arguments of the Normal-Gamma prior, as shown in Section 4.3.2.

```
R> params_uni <- list(a = 1, b = 1, c = 1, prior_var_phi = 0.1,
+                     prior_delta_c = 1, prior_delta_d = 1)
```

The algorithm is run by calling function `detect_cp`. We include the list `params_uni` in the argument `params` and we also specify other general arguments, common to both the time series and epidemiological frameworks. Specifically, we specify the number of iterations, the number of burn-in steps and the probability of performing a split at each iteration.

```
R> out <- detect_cp(data, n_iterations = 10000, n_burnin = 5000, q = 0.25,
+           params = params_uni, kernel = "ts")
R> print(out)

DetectCpObj object
Type: change point detection on univariate time series
```

To get a posterior estimate of the change points we use `posterior_estimate`, we leave all arguments to default values and set *Binder* as loss function.

```
R> cp_est <- posterior_estimate(out, loss = "binder")
```

The output of `posterior_estimate` is a sequence of number that represents the allocation of each realisation to a block. In order to get the position of the change points it is sufficient to print the cumulative sum of the frequency table of the vector, remove the last element and add one.

```
R> cumsum(table(cp_est))[-length(table(cp_est))]

  1   2
 51 151
```

Finally we graphically represent the detected change points along with the time series with `plot`. This method also provides, by setting `plot_freq = TRUE`, the histogram with the frequency of times that each realisation has been detected as change point in the MCMC chain. In Figure 4.3 is reported the output generated by the following code

```
R> plot(x = out, loss = "binder", plot_freq = TRUE)
```

In the multivariate scenario, we need to define a matrix in which each row corresponds to a dimension of the time series. We consider the same number of realisations and the same change point locations as in the univariate example. A synthetic multivariate time series is generated with the following code:

```
R> data <- matrix(NA, nrow = 3, ncol = 200)
R> data[1, 1] <- rnorm(1, mean = 1.20, sd = 0.12)
R> data[2, 1] <- rnorm(1, mean = 1.15, sd = 0.15)
```

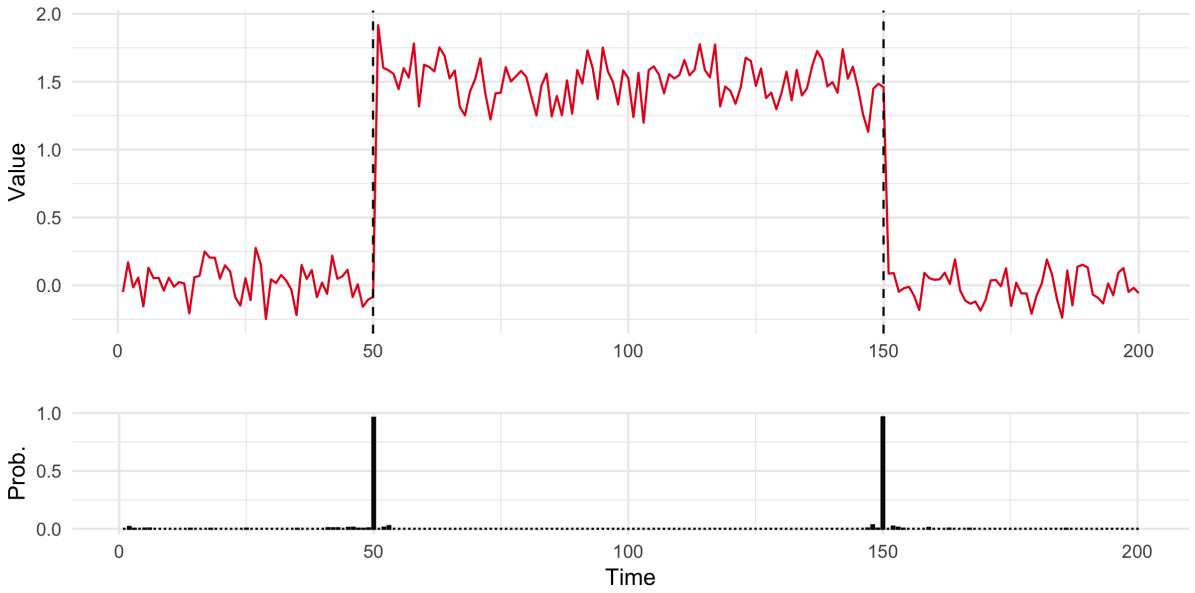


Figure 4.3: Detected change points on univariate synthetic time series. The dashed lines represent the estimated change points.

```
R> data[3, 1] <- rnorm(1, mean = 1.10, sd = 0.14)
R> for(i in 2:50){
R>  data[1, i] <- 0.1 * data[1, i-1] + (1 - 0.1) * 1.20 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.12)
R>  data[2, i] <- 0.1 * data[2, i-1] + (1 - 0.1) * 1.15 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.15)
R>  data[3, i] <- 0.1 * data[3, i-1] + (1 - 0.1) * 1.10 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.14)
R> }
R> data[1, 51] <- rnorm(1, mean = 0.06, sd = 0.14)
R> data[2, 51] <- rnorm(1, mean = 0.07, sd = 0.12)
R> data[3, 51] <- rnorm(1 ,mean = 0.08, sd = 0.10)
R> for(i in 52:150){
R>  data[1, i] <- 0.1 * data[1, i-1] + (1 - 0.1) * 0.06 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.14)
R>  data[2, i] <- 0.1 * data[2, i-1] + (1 - 0.1) * 0.07 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.12)
R>  data[3, i] <- 0.1 * data[3, i-1] + (1 - 0.1) * 0.08 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.10)
R> }
R> data[1, 151] <- rnorm(1, mean = 0.72, sd = 0.13)
R> data[2, 151] <- rnorm(1, mean = 0.69, sd = 0.10)
R> data[3, 151] <- rnorm(1, mean = 0.75, sd = 0.14)
R> for(i in 152:200){
R>  data[1, i] <- 0.1 * data[1, i-1] + (1 - 0.1) * 0.72 +
```

```
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.13)
R>  data[2, i] <- 0.1 * data[2, i-1] + (1 - 0.1) * 0.69 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.10)
R>  data[3, i] <- 0.1 * data[3, i-1] + (1 - 0.1) * 0.75 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.14)
R> }
```

Similarly to the previous case, we need to specify the parameters of the Normal-Inverse Wishart distribution.

```
R> params_multi <- list(m_0 = rep(0, 3), k_0 = 1, nu_0 = 5,
+                       S_0 = diag(0.1, 3, 3), prior_var_phi = 0.1,
+                       prior_delta_c = 1, prior_delta_d = 1)
```

We run the algorithm providing `params_multi` as argument, specifying also other parameters as in the univariate scenario.

```
R> out <- detect_cp(data, n_iterations = 10000, n_burnin = 5000, q = 0.5,
+           params = params_multi, kernel = "ts")
R> print(out)

DetectCpObj object
Type: change point detection on multivariate time series
```

Figure 4.4 shows the graphical representation of the estimated change points with the *Binder loss function*, along with the observed data and the probability of having a change in the time domain. Each color corresponds to a specific dimension.

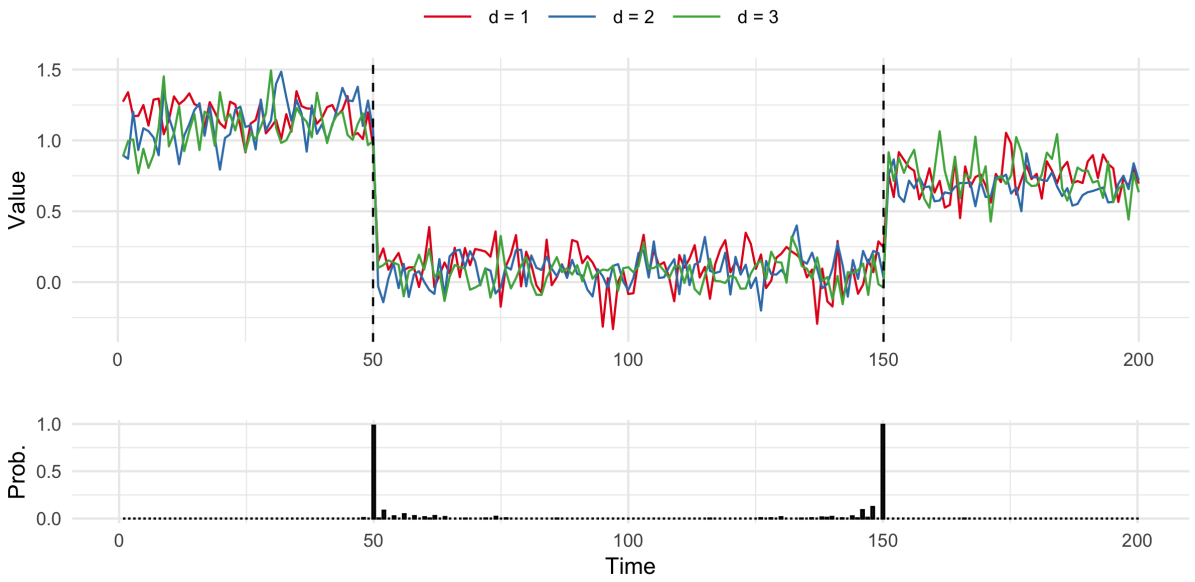


Figure 4.4: Detected change points on multivariate synthetic time series. The dashed lines represent the position of the estimated change points. Each color denotes a different dimension of the time series.

To show how to detect change points on a survival function, we start by generating synthetic infection times on an interval of time $(0, 200)$. We assume a population of 10 000 individuals, of which 50 are infected at time 0. The recovery rate is set at 1/8 and a change point occurs at time 131, where the infection rate switches from 0.2 to 0.55. With the following code we create the vector of infection rates

```
R> betas <- c(rep(0.2, 130), rep(0.55, 70))
```

The function `sim_epi_data` returns a vector of continuous infection times. Since the input of `detect_cp` requires discrete time points, we round the output of `sim_epi_data` and compute the number of new infections at each time. The result is a one-column matrix, `inf_count`, where each entry represents the number of new infections at the corresponding time indicated by the row.

```
R> inf_times <- sim_epi_data(S0 = 10000, I0 = 50, max_time = 200,
+                 beta_vec = betas, xi_0 = 1/8)
R> inf_times <- table(floor(inf_times))
R> inf_count <- matrix(0, 1, 200)
R> inf_count[as.numeric(names(inf_times)), 1] <- inf_times
```

After specifying the specific parameters of the survival function on list `params_epi`, we run the algorithm.

```
R> out <- detect_cp(data = inf_count, n_iterations = 5000, n_burnin = 2500,
+           q = 0.25, params = params_epi, kernel = "epi")
R> print(out)

DetectCpObj object
Type: change point detection on an epidemic diffusion
```

Figure 4.5 shows the output obtained with method `plot`, the change points are shown on the empirical survival function of the data generating infection times model.

4.4.2 Cluster time dependent data with common change points

Function `clust_cp` cluster time series or epidemic diffusions with common change points. The wrapper will perform the proper algorithm, depending on the type of input data. First, we cluster univariate time series. Data are in the form of a matrix, where each row is a time series and each column a time instant. We consider 5 time series with 200 realisations each, where observations are divided in two groups, of size 3 and 2 respectively. For the first group, we assume that data share two change points, one at time 51 and one at time 151. For the second group, there is only one change point at time 26. The code for generating these data is reported in Appendix C.2. To run the algorithm, it is mandatory to specify which kind of kernel describes the data, here `kernel = "ts"`. We specify in a list the required tuning parameters, which are the same of univariate change point detection, except for the autoregressive coefficient ϕ, here assumed to be fixed.

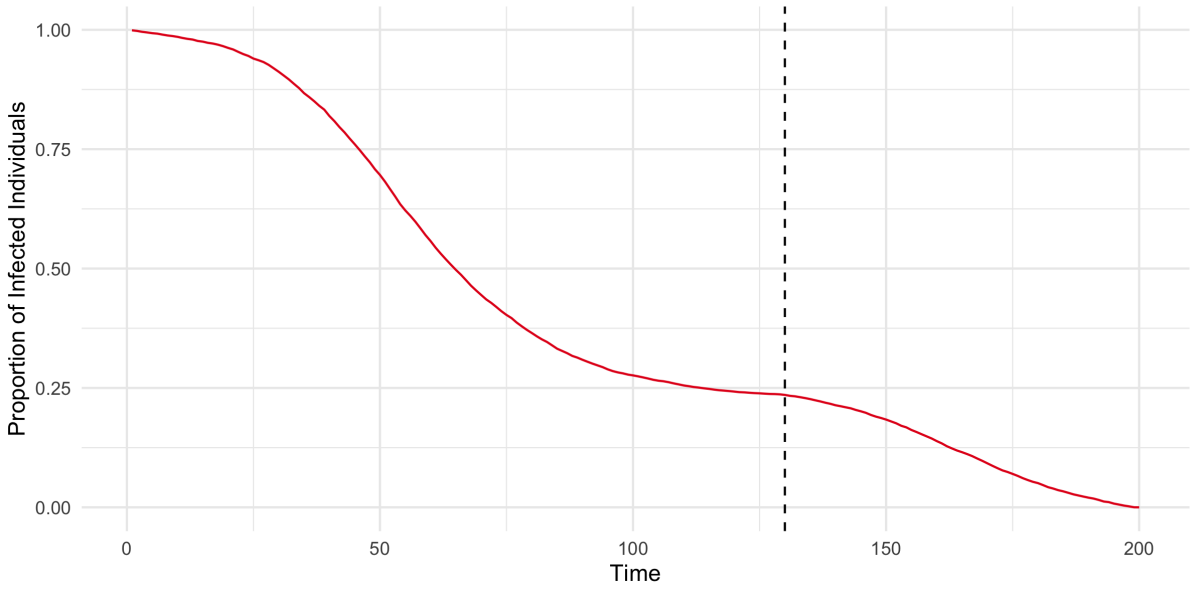


Figure 4.5: Detected change points on a synthetic epidemiological diffusion. The dashed line represents the position of the change point estimated by the model.

```
R> params_uni <- list(a = 0.1, b = 1, c = 1, phi = 0.1)
```

We run the algorithm with the following code.

```
R> out <- clust_cp(data, n_iterations = 10000, n_burnin = 5000,
+                  L = 1, q = 0.5, B = 10000, params = params_uni,
+                  kernel = "ts")
R> print(out)

ClustCpObj object
Type: clustering univariate time series with common change points
```

As shown by the `print` method, the algorithm detects that time series are univariate. We estimate the latent partition of the data by calling the `posterior_estimate` function. The output is a vector of numbers denoting the clustering allocation of each observation.

```
R> posterior_estimate(out, loss = "binder)
R> plot(out, loss = "binder)

[1] 1 1 1 2 2
```

The method `plot` shows a graphical representation of the estimated latent partition of the data, which is shown in Figure 4.6. Different colors denote different observations, while different line types denote different clusters.

```
R> plot(out, loss = "binder)
```

When considering multivariate time series, each observation is a multivariate time series, we need to define data as an array. Each slice of the array, denoted as the third index, is a matrix

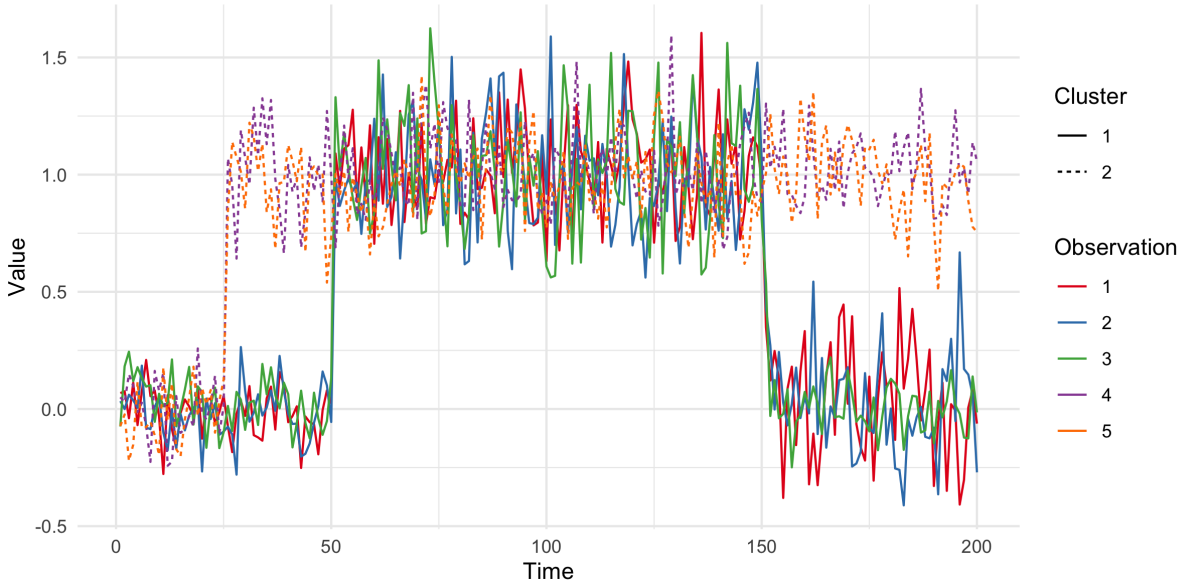


Figure 4.6: Clustering univariate time series with common change points. Different colors denote observations, different line types denote the cluster assignments.

that corresponds to an observation. For each observation, the number of rows corresponds to the dimensions of the time series, while the number of columns is the number of observational times. Here we sample 5 time series, each one observed at 200 times and with 2 dimensions. The code to generate data is available in Appendix C.2. We create a list specifying the parameters for the multivariate kernel function.

```
R> params_multi <- list(m_0 = rep(0,2), k_0 = 1, nu_0 = 5,
+                       S_0 = diag(1, 2, 2), phi = 0.1)
```

We run the algorithm calling the `detect_cp` function. Figure 4.7 shows a graphical representation of the posterior point estimate of the data clustering.

```
R> out <- clust_cp(data = data, n_iterations = 10000, n_burnin = 5000,
+                  L = 1, B = 10000, params = params_multi, kernel = "ts")
R> print(out)

ClustCpObj object
Type: clustering multivariate time series with common change points
```

Finally, we show how to apply `clust_cp` to epidemic diffusions. Input data are of matrix form, where rows denote different populations and columns the observational times. Each entry is the number of new infected individuals at a specific time in a specific population. We generate infection times from 3 populations and 50 time instants. We assume two groups, the first with two observations and a change point at time 25 and the second with one observation and a change point at time 10. To generate these data, we use `sim_epi_data`. For each population, we consider 10 000 individuals, of which 20 are infected at time 0, and a recovery rate equal to 1/8.

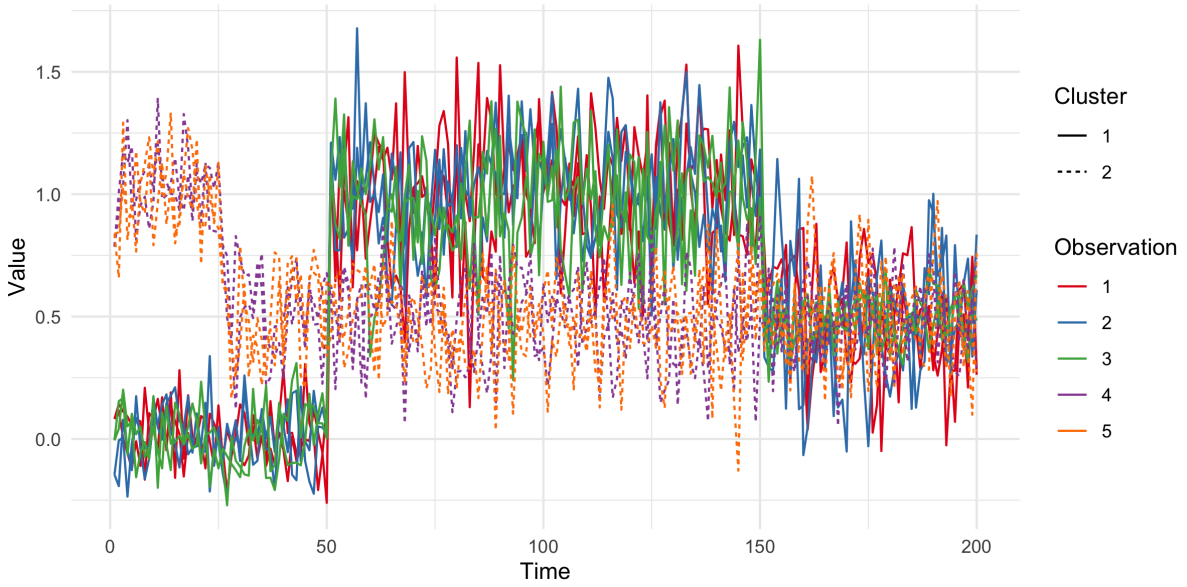


Figure 4.7: Clustering multivariate time series with common change points. Different color denote observations, different line types denote the cluster assignments.

```
R>  data <- matrix(0, nrow = 3, ncol = 200)
R>  inf_times <- list()
R>
R>  betas <- list(c(rep(0.211, 120),rep(0.55, 80)),
+                 c(rep(0.215, 120),rep(0.52, 80)),
+                 c(rep(0.193, 30),rep(0.53, 170)))
R>
R> for(i in 1:3){
R>  inf_times[[i]] <- sim_epi_data(10000, 20, 200, betas[[i]], 1/8)
R>  inf_times[[i]] <- table(floor(inf_times[[i]]))
R>  data[i,as.numeric(names(inf_times[[i]]))] <- inf_times[[i]]
R> }
```

We run the `clust_cp` function for epidemic diffusions. Specifically, we set `kernel = "epi"`. Similarly to before, we create a list with the specific parameters of the algorithm. Specifically, the number of Monte Carlo iterations for the likelihood integration, the recovery rate, the parameters of the weights distribution, the parameters of the gamma and the normal proposals, and the average number of blocks when a latent order is randomly generated.

```
R> params_epi <- list(M = 1000, xi = 1/8, alpha_SM = 1,
+                     a0 = 3, b0 = 10, I0_var = 0.1, avg_blk = 5)
R> out <- clust_cp(data, n_iterations = 5000, n_burnin = 2000,
+                  L = 1, B = 1000, params = params_epi, kernel = "epi")
R> print(out)

ClustCpObj object
Type: clustering epidemic diffusions with common change points
```

Figure 4.8 shows the posterior point estimate of the clusters along with the empirical survival functions of the epidemic diffusions, specific for each population.

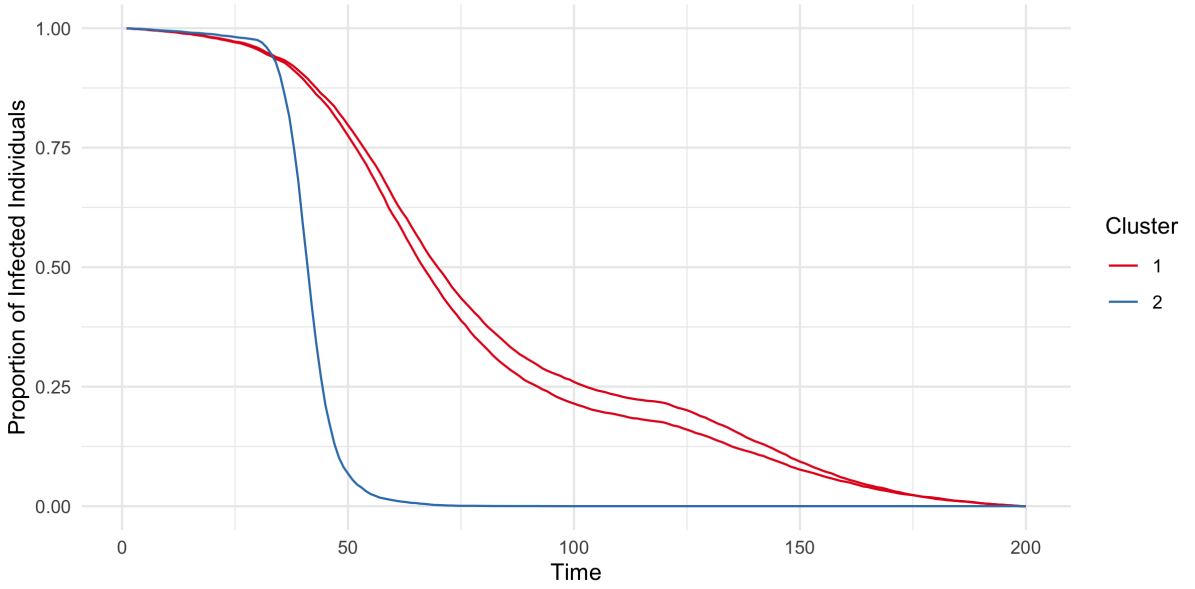


Figure 4.8: Clustering of survival functions with common change points. Different colors denote the cluster assignments.

4.5 Summary and discussion

We presented in this chapter **BayesChange**, an R package written in C++ that provides Bayesian methods for change point analysis. The package offers two key contributions: (1) a function for detecting change points on time series and epidemic diffusions, which is not available in other R packages, and (2) the implementation of a novel method to cluster time-dependent data sharing common change points. The R interface makes **BayesChange** accessible to non-advanced users, while the underlying C++ implementation ensures computational efficiency.

4.6 Computational details

The results in this chapter were obtained using R 4.4.3 with the **BayesChange** 2.1.1 package on a macOS 15.2 machine with Apple M3 chip and the dependencies of the following packages: **Rcpp** 1.0.14, **RcppArmadillo** 14.4.31, **RcppGSL** 0.3.13, **salso** 0.3.53, **dplyr** 1.1.4, **tidyr** 1.3.1, **ggplot2** 3.5.2, and **ggpubr** 0.6.0. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Chapter 5

Repulsive Clustering of Time Series Sharing Asynchronous Change Points

5.1 Introduction

In this project, we propose a model that cluster together time series with common change points, with the assumption that the locations of the change points in two time series that belong to the same cluster must not be exactly the same, but may occur in close proximity. Consider, for example, the standardized and detrended average between the opening and closing daily stock price of two companies, UnitedHealth Group and Verizon Communications, from 2020-01-01 to 2022-01-02, shown in Figure 5.1. When computing the change points of the two stock prices marginally, we observe that they are very close. However, if we cluster the two time series according to the locations of their change points, they are assigned to different clusters. We aim to build a model that clusters time series of this kind together. Although the locations of their change points differ, they can be considered as equivalent and their differences due to some noise that naturally arises in real data. In literature there are few works that perform change point detection assuming that change points on different time series might occur close to each other and not exactly at the same moment. For example, the works by Hallgren et al. (2024) and McKee and Kalli (2025) propose to detect asynchronous changes with a Bayesian approach where the dependency between the change points is modeled through a latent directed graph. In the frequentist context this problem is addressed by Xie et al. (2019) that propose a model where the delay of the change points is fixed or can be eliminated by synchronizing the sensors that collect the time series and by Fisch et al. (2022) that instead consider fixed windows where the asynchronous change points might be included. In the context of clustering time series with common change points, we are inspired by the model introduced in Chapter 3 where the time series that belong to the same cluster share the same latent order induced by the change points. The prior distribution of this latent order is a finite mixture where the atoms correspond to all the possible latent orders (see Chapter 2 for further details). Following this method, we propose to redefine the prior distribution of the latent orders, by restricting the mixture to a fixed number of components, inducing repulsiveness among the atoms. Hence, the latent orders differ from each other and we possibly cluster together time series with asynchronous change points. Early

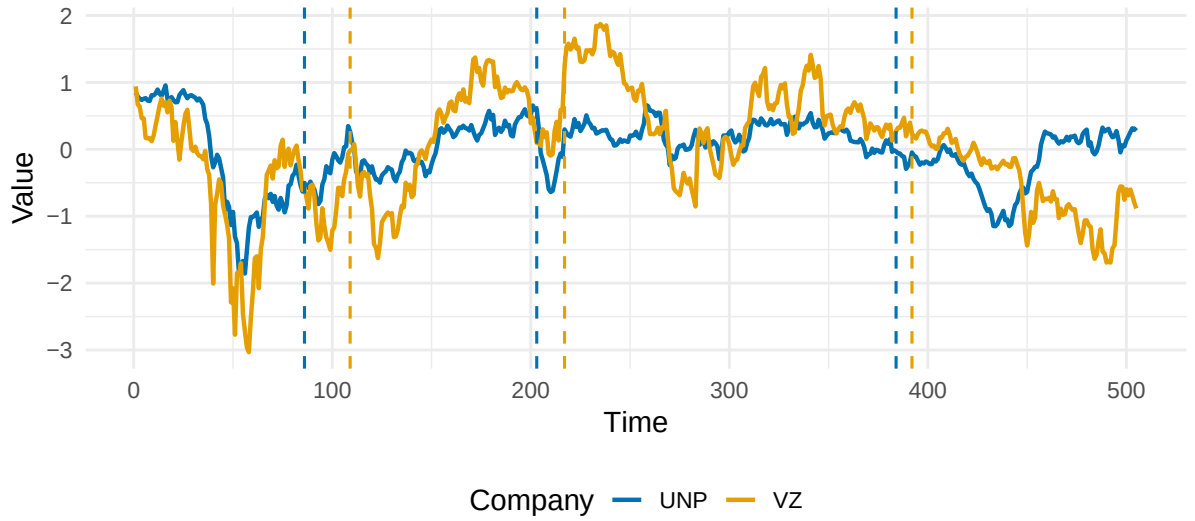


Figure 5.1: Time series of the average between the opening and closing daily stock prices of UnitedHealth Group and Verizon Communications from 2020-01-01 to 2022-01-02. Dashed lines represent the marginal change points of the two time series estimated with the model by Martínez and Mena (2014).

contributions on Bayesian repulsive mixture models can be found in Petralia et al. (2012), where the authors propose a prior for the atoms that incorporates a repulsive function. This function depends on the distance between two atoms. If an atom is too close to the other components of the mixture, its sampling probability will be reduced. This method was later studied in the context of density regression also by Quinlan et al. (2018) and Quinlan et al. (2021). We base our proposal on these approaches. Other recent approaches on Bayesian repulsive mixtures are based on determinantal point processes (Lavancier et al., 2015), see for example Xu et al. (2016), Bianchini et al. (2020), Beraha et al. (2022) and Ghilotti et al. (2024). These methods are a starting point for our problem, however applying determinantal point processes to discrete data appears to be a challenging task. This chapter is structured as follows: Section 5.2 presents the model we use with a particular focus on the repulsive structure we implement on the new prior distribution for the latent orders. In Section 5.3 is presented the posterior procedure with details on the algorithm, Section 5.4 is an extensive simulation study and Section 5.5 shows preliminary results on financial real data.

5.2 Model

Similarly to Section 3.2, we consider a set of n discrete time series $\mathcal{Y} = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_n\}$, where a generic time series y_i is a set of T realisations $\boldsymbol{y}_i = (y_{i,1}, \dots, y_{i,T})$ where $y_{i,t} \in \mathbb{R}$. We assume that each realisation is distributed according to a parametric distribution f,

$$y_{i,t} \sim f(\cdot \mid \theta_{i,t}), \tag{5.1}$$

where $\theta_{i,t}$ is the parameter of f for observation i at time t. A change point occurs whenever this parameter changes, so if $\theta_{i,t} \neq \theta_{i,t+1}$ then a change point occurs for observation $\boldsymbol{y}_i$ at time t. Depending on the distributional form of f, $\theta_{i,t}$ might be univariate or multivariate. Following the same approach of Chapter 3, we consider the order induced by $k_i - 1$ change points, $\rho_i = (A_{i,1}, \ldots, A_{i,k_i})$, as a random variable distributed according to a Dirichlet-Multinomial distribution. Our aim is to obtain a point estimate of $\lambda = \{B_1, \ldots, B_k\}$, a partition of $\mathcal{Y}$ where two observations that belong to the same cluster B_j if they share the same change points. More formally, $B_j = \{i : \rho_i = \rho^{\dagger}_{(j)}\}$ with $\{\rho^{\dagger}_{(1)}, \ldots, \rho^{\dagger}_{(k)}\}$ denoting the set of unique latent orders associated to clusters in λ.

5.2.1 Repulsiveness among latent orders

To perform clustering of time series with similar, but not necessarily equal change points, we design a method where the latent order ρ_i of a generic time series y_i is sampled from a mixture model with a fixed number M of components,

$$p(\rho) = \sum_{j=1}^{M} \pi_j \delta_{\widetilde{\rho}_j}(\rho), \tag{5.2}$$

where the atoms are M combinations of T realisations in a random number of groups. To avoid groups with very similar latent orders, we need the M components of (5.2) to be sufficiently heterogeneous. To do so we induce repulsiveness in the sampling procedure of $\widetilde{\rho}_1, \ldots, \widetilde{\rho}_M$, so that it does not happen that different components have very similar latent orders.

To introduce repulsiveness among the atoms in Equation (5.2) we resort to the strategy proposed by Petralia et al. (2012), a seminal work in the context of Bayesian repulsive mixture models. The authors propose as joint prior for the random orders $\widetilde{\rho} = \{\widetilde{\rho}_1, \ldots, \widetilde{\rho}_M\}$, that acts as atoms in (5.2), the following distribution:

$$p(\widetilde{\rho}) = C_1 \left(\prod_{j=1}^{M} g_0(\widetilde{\rho}_j) \right) h(\widetilde{\rho}), \tag{5.3}$$

where $g_0(\widetilde{\rho}_j)$ is the prior distribution from which a generic latent order is sampled, $h(\widetilde{\rho})$ is the repulsive function based on a distance $d(\cdot, \cdot)$ and C_1 the normalisation constant. We choose as repulsive function among the latent orders the following

$$h(\widetilde{\rho}) = \prod_{(s,j) \in A} g\{d(\widetilde{\rho}_s, \widetilde{\rho}_j)\}, \tag{5.4}$$

where $A = \{(s, j) : s = 1, \ldots, M; j < s\}$ and $g : \mathbb{R}_+ \to [0, U]$ is a strictly monotone differentiable function with $g(0) = 0$, $g(x) > 0$ for all $x > 0$ and $U < \infty$. As function g again we resort on the proposal by Petralia et al. (2012):

$$g\{d(\rho_s, \rho_j)\} = \exp\left[-\tau \{d(\rho_s, \rho_j)\}^{-\nu}\right], \tag{5.5}$$

where $\tau > 0$ denotes the scale parameter, and $\nu \in \mathbb{Z}^+$ determines how quickly the function g

approaches zero as the distance $d(\rho_s, \rho_j)$ becomes smaller. It should be noted that also $\tau = 0$ is admitted and corresponds the non repulsive case. Since in Petralia et al. (2012) the mixture model is designed for real valued data, and in our model data are discrete objects, we need to choose as distance a metric that computes the dissimilarity of two partitions. The literature is very broad in this context, we decide to resort to variation of information (Meilă, 2007) because for our purpose it has a more reasonable behaviour compared to other measures like the Binder loss function (Binder, 1978). For example if we consider a latent order with 99 change points $\rho_1 = \{1, \dots, 1\}$, one with 2 change points $\rho_2 = \{45, 55\}$ and one with no change points $\rho_3 = \{100\}$ we have that the distance between these orders in terms of variation of information is

$$VI(\rho_1, \rho_2) = 5.651082 \quad VI(\rho_2, \rho_3) = 0.9927745,$$

while in terms of Binder loss function is

$$BI(\rho_1, \rho_2) = 0.495 \quad BI(\rho_2, \rho_3) = 0.495.$$

According to the Binder loss function, a latent order with two change points is equally distanced from one with zero change points and one with the maximum possible number of change points. On the other hand the variation of information considers the order with two change points much closer to the one with zero change points then to the one with the maximum number of changes.

5.2.2 Model specification

Since our model is an extension of the one described in Chapter 3, we assume the same hierarchical modeling structure with the addition of the repulsive mixture presented earlier. The distribution of the generic $\boldsymbol{y}_i$ is conditioned on its latent order ρ_i and on the set of local parameters $\boldsymbol{\theta}_i^*$. We assume a Markovian dependency structure for the observations inside each regime of the time series: observation t then depends on the realisation at time $t-1$ and on the corresponding parameter $\theta_{i,t}$. The weights of mixture (5.2) are assumed to be distributed as a Dirichlet distribution with parameters $\alpha_1 = \alpha_2 = \cdots = \alpha_M = 1$. Finally a prior $P_0(\theta)$ is set for the unique parameters of each block. The full hierarchical model has the following form

$$\begin{aligned}
\boldsymbol{y}_i \mid \rho_i, \boldsymbol{\theta}_i^* &\sim \prod_{j=1}^{k_i} \prod_{t=t_{i,j}^-}^{t_{i,j}^+} \mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta_{i,j}^*), \quad i = 1, \dots, n, \\
\rho_i \mid \widetilde{p}(\cdot) &\overset{\text{iid}}{\sim} \widetilde{p}(\cdot) = \sum_{j=1}^{M} \pi_j \delta_{\widetilde{\rho}_j}, \quad i = 1, \dots, n, \\
(\widetilde{\rho}_1, \dots, \widetilde{\rho}_M) &\sim \pi(\widetilde{\rho}) = C_1 \left(\prod_{j=1}^{M} g_0(\widetilde{\rho}_j) \right) h(\widetilde{\rho}) \\
(\pi_1, \dots, \pi_M) &\sim \text{DIR}(\alpha_1, \dots, \alpha_M), \\
\theta_{i,j}^* &\overset{\text{iid}}{\sim} P_0(\theta), \quad j = 1, \dots, k_i, \; i = 1, \dots, n,
\end{aligned} \tag{5.6}$$

where $t^-_{i,j}$ and $t^+_{i,j}$ are respectively the first and last index of block j for time series $\boldsymbol{y}_i$. The law $\mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta^*_{i,j})$ is derived from an Ornstein-Uhlenbeck process, as done in Chapter 2.

5.2.3 Prior properties

To study the properties of the prior distribution, we analyze some samples from (5.3) by running the procedure depicted in Algorithm 5. We define g_0 using a hierarchical Binomial-Multinomial distribution. In this setup, the number of groups in a latent order is first drawn from a Binomial distribution with parameter $p \in (0, 1)$. Conditioned on this number, the cardinalities of the groups are determined via a Multinomial distribution. We try different scenarios with different values of M and τ. As suggested by Petralia et al. (2012), we set $\nu = 2$ for all configurations. Figure 5.2 shows the boxplots of the variation of information (VI) distances among the M latent orders sampled with different values of M and τ. For each configuration we run the sampling procedure for 1 000 iterations. Figure 5.2 clearly shows that as τ increases, the repulsiveness among the orders also increases. Moreover, we can notice that with a larger value of τ the number of outliers increases. Also increasing the number of elements M and keeping τ constant the number of outliers increases, but the average distance between the orders does not change.

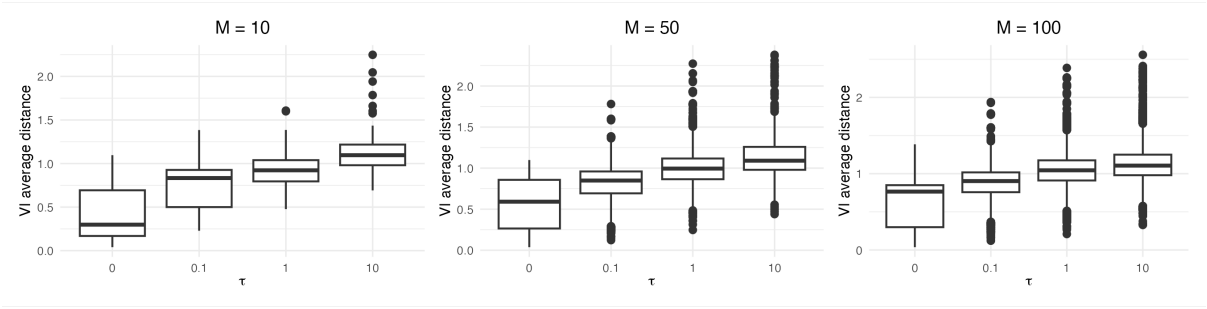


Figure 5.2: distribution of the distance among samples of random orders from the prior distribution (5.3) with different number of components M and the degree of repulsiveness τ.

5.3 Posterior sampling

The sampling procedure that we implement is inspired by the Metropolis-within-Gibbs algorithm for repulsive mixtures introduced by Quinlan et al. (2021). While in the algorithm by Quinlan et al. (2021) the atoms of the mixture components are real values, in our model the atoms are time-ordered partitions. Our main challenge then consists in building a proper procedure to update the atoms of the mixture. In the burn-in steps the proposed orders for the M components are sampled from a Binomial-Multinomial scheme. According to this procedure, we firstly sample the number of groups in the order (i.e. the number of regimes induced by the change points) from a Binomial distribution and then the frequency of observations in each regime is sampled from a Multinomial distribution. The probability of a success in the Binomial distribution is set as the number of expected regimes divided by the number of time realisations. By doing so we can set a sort of prior guess on the number of change points. In the Multinomial distribution we

want all the blocks of the order to have the same prior probability, so we set all the parameters equal to 1 divided by the number of blocks. In the main part of the algorithm, to update the atoms of the mixture, we implement instead a split and merge procedure inspired by the method of Martínez and Mena (2014) for change point detection. Whenever we propose a new order for a component, we start from the previous order and randomly pick a block that we split with probability q or we merge with the subsequent one with probability $1-q$. With this procedure, during the burn-in phase, new orders are proposed by exploring more broadly the space of possible orders, allowing for proposals that might differ substantially from the previous one. On the other hand, during the main phase, since we perform a split or a merge of the previous order, the proposed order lies in the local neighborhood of the current one.

5.3.1 Posterior sampling algorithm

Here we present the algorithm to sample from the posterior distribution. The algorithm is divided in three phases: a start phase, where the initial values are set, the burn-in phase and the main phase.

Start: set the following initial values:

- $z_i^{(0)} = i$ for $i = 1, \dots, n$
- $\pi_{M,1}^{(0)} \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_M)$
- sample $\rho_j^{(0)} = (n_j^{(0)}, \dots, n_{m_j^{(0)}}^{(0)})$ for $j = 1, \dots, M$ according to the following procedure:

  i) sample $m_j^{(0)} \sim \text{Binomial}(T-1, p)$

  ii) sample $\rho_j^{(0)} \sim \text{Multinomial}\left(T, \left(\frac{1}{m_j^{(0)}}, \dots, \frac{1}{m_j^{(0)}}\right)\right)$

Burn-in: repeat for B times the following scheme:

1. for $i = 1, \dots, n$

   1.1 compute $\pi_j^{(t,i)} = \frac{\pi_j^{(t)} \mathcal{L}(\rho_j^{(t)} | y_i)}{\sum_{l=1}^{M} \pi_l^{(t)} \mathcal{L}(\rho_l^{(t)} | y_i)}$

   1.2 get $z_i^{(t+1)} \sim \text{Multinomial}\left(1, \pi_{M,1}^{(t,i)}\right)$

2. $\pi_{M,1}^{(t)} \sim \text{Dirichlet}(\alpha_{M,1}^{(t)})$, where

   - $\alpha_{M,1}^{(t)} = \left(\alpha_1 + n_1^{(t+1)}, \dots, \alpha_M + n_M^{(t+1)}\right)$
   - $n_j^{(t+1)} = \text{card}(\{i : z_i^{(t+1)} = j\})$

3. for $j = 1, \dots, M$

   3.1 generate a candidate ρ_j^* with the following scheme

   - sample $k_j^* \sim \text{Binomial}(p)$
   - sample $\rho_j^* \sim \text{Multinomial}\left(\frac{1}{k_j^*}, \dots, \frac{1}{k_j^*}\right)$

3.2 update $\rho_j^{(t+1)} = \rho_j^*$ with probability $\min(1, \beta_j)$, where

- $\beta_j = \dfrac{\prod\limits_S \mathcal{L}(y_i|\rho_j^*)}{\prod\limits_S \mathcal{L}(y_i|\rho_j^{(t)})} \dfrac{\binom{T}{k_j^*} p^{k_j^*}(1-p)^{T-k_j^*}}{\binom{T}{k_j^{(t)}} p^{k_j^{(t)}}(1-p)^{T-k_j^{(t)}}} \dfrac{T! \prod_{i=1}^{m_j^*} ((k_j^*)^{m_{j,i}^*})^{-1}}{T! \prod_{i=1}^{k_j^{(t)}} ((k_j^{(t)})^{m_{j,i}^{(t)}})^{-1}} \overline{\omega}_j$
- $\overline{\omega}_j = \prod\limits_{\{l \in [M] \setminus \{j\}\}} \dfrac{g(d(\rho_j^*, \rho_l^{(t)}))}{g(d(\rho_j^{(t)}, \rho_l^{(t)}))} = \prod\limits_{\{l \in [M] \setminus \{j\}\}} \dfrac{exp(-\tau\{d(\rho_j^*, \rho_l^{(t)})\}^{-\nu})}{exp(-\tau\{d(\rho_j^{(t)}, \rho_l^{(t)})\}^{-\nu})}$

and $S = \{y_i : z_i^{(t+1)} = j\}$

Main: same as burn-in phase, but change step 3 with the following split and merge scheme

- With probability $q\mathbb{1}_{[1<k_j^{(t)}<T]} + \mathbb{1}_{[k_j^{(t)}=T]}$ perform a split:
  - choose h from $\{h : 1 \le h \le k_j^{(t)} : m_{j,h}^{(t)} > 1\}$
  - choose l from $\{1, \dots, m_{j,h}^{(t)}\}$
  - set $\rho_j^* = (m_{j,1}^{(t)}, \dots, l, m_{j,h}^{(t)} - l, \dots, m_{j,k_j^{(t)}}^{(t)})$
  - update $\rho_j^{(t+1)} = \rho_j^*$ with probability $\min(1, \beta_j)$, where
  - set $S = \{y_i \mid z_i^{(t+1)} = j\}$
    - $\beta_j = \begin{cases} \dfrac{\prod\limits_S \mathcal{L}\left(y_i \mid \rho_j^*\right)}{\prod\limits_S \mathcal{L}\left(y_i \mid \rho_j^{(t)}\right)} \cdot \dfrac{1-q}{q} \cdot \dfrac{\sum\limits_{l=1}^{k_j^{(t)}} \mathbb{1}_{[m_{j,l}>1]}(m_{j,l}-1)}{m_j^{(t)}} \cdot \bar{\omega}_j & \text{if } 1 \le M \le T \\ \dfrac{\prod\limits_S \mathcal{L}\left(y_i \mid p_j^*\right)}{\prod\limits_S \mathcal{L}\left(y_i \mid p_j^{(t)}\right)} \cdot (1-q)(T-1) \cdot \bar{\omega}_j & \text{if } M = T \end{cases}$
    - $\overline{\omega}_j$ = see 3.2 in the burn-in step.

- With probability $(1-q)\mathbb{1}_{[1<k_j^{(t)}<T]}$ perform a merge:
  - choose h from $\{1, \dots, k_j^{(t)} - 1\}$
  - set $\rho_j^* = (m_{j,1}^{(t)}, \dots, m_{j,h}^{(t)} + m_{j,h+1}^{(t)}, \dots, m_{j,m_j^{(t)}}^{(t)})$
  - set $S = \{y_i \mid z_i^{(t+1)} = j\}$
  - accept $\rho_j^{(t+1)} = \rho_j^*$ with probability $\min(1, \beta_j)$ where
    - $\beta_j = \begin{cases} \dfrac{\prod\limits_S \mathcal{L}\left(y_i \mid \rho_j^*\right)}{\prod\limits_S \mathcal{L}\left(y_i \mid \rho_j^{(t)}\right)} \cdot \dfrac{q}{1-q} \cdot \dfrac{M-1}{\sum\limits_{l=1}^{k_j^*} \mathbb{1}_{[m_{j,l}^*>1]} \left(m_{j,l}^{(t)} + m_{j,l+1}^{(t)} - 1\right)} \cdot \bar{\omega}_j & \text{if } 1 < k_j^* < T \\ \dfrac{\prod\limits_S \mathcal{L}\left(y_i \mid \rho_j^*\right)}{\prod\limits_S \mathcal{L}\left(y_i \mid \rho_j^{(t)}\right)} \cdot q \cdot (T-1) \cdot \bar{\omega}_j & \text{if } k_j^* = T \end{cases}$
    - $\overline{\omega}_j$ = see 3.2 in the burn-in step.

5.4 Simulation Study

In order to assess the functioning of the algorithm, and to study the properties of the posterior distribution, we propose a simulation study on synthetic data.

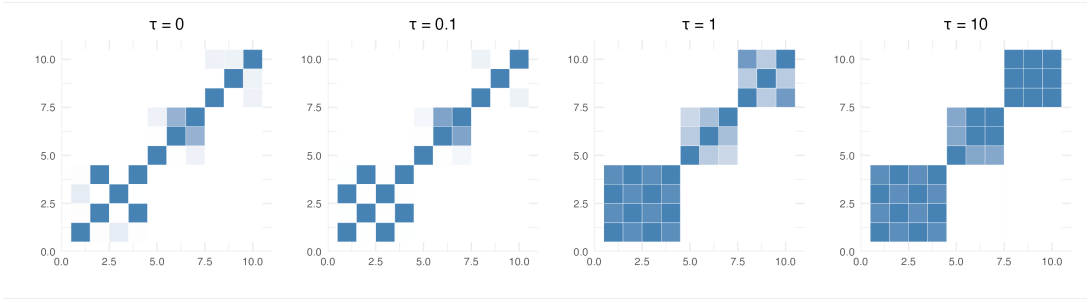
We consider a simulation setting in which time series within the same cluster exhibit similar, but not necessarily identical, change points. Specifically, we generate $n = 10$ time series, each consisting of $T = 300$ observations, according to the model in Equation 3.9. The data are partitioned into $k = 3$ clusters, such that time series within each cluster share change points that may either coincide exactly or differ by a small temporal offset. In particular, we allow the positions of corresponding change points to differ by at least 5 and at most 20 time points, in order to assess the robustness of the model when change points are close but not perfectly aligned. To further increase the complexity of the simulation setting, we assume that time series belonging to the same cluster do not share the same regime-specific parameters and we allow both the mean and the variance to change at each change point.

The parameters used for data generation are summarized in Table 5.1. We fix $\nu = 2$ and investigate the performance of the algorithm across different configurations of $\tau \in \{0, 0.1, 1, 10\}$ and $M \in \{10, 50, 100\}$. Each configuration is evaluated over 50 independent replications of the data and run for 5 000 iterations of which 1 000 are discarded as burn-in.

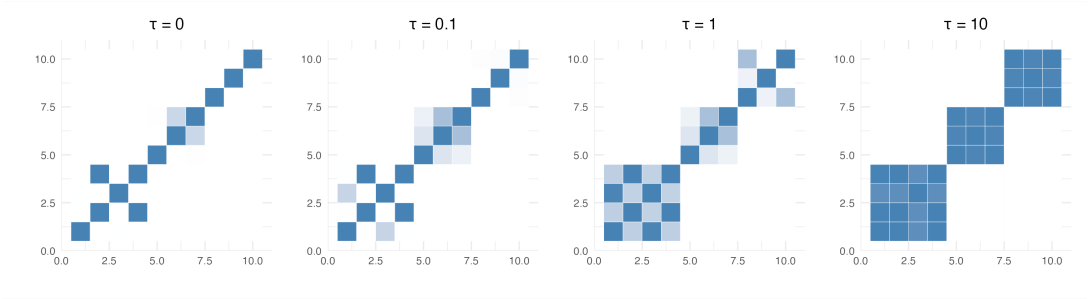
i	j	$(m_{i,1}, \dots, m_{i,k_i})$	$(\mu^*_{i,1}, \dots, \mu^*_{i,k_i})$	$(\eta^*_{i,1}, \dots, \eta^*_{i,k_i})$
1	1	{51, 249}	{0.25, 0.75}	{0.0081, 0.0081}
2	1	{40, 260}	{0.15, 0.65}	{0.0119, 0.0182}
3	1	{60, 240}	{0.45, -0.05}	{0.0081, 0.0115}
4	1	{40, 260}	{0.5, -0.2}	{0.0081, 0.0115}
5	2	{140, 119, 41}	{-0.5, 0.25, -0.15}	{0.0132, 0.0081, 0.0065}
6	2	{135, 114, 51}	{0.85, 0.15, 1.25}	{0.0132, 0.0065, 0.0115}
7	2	{135, 129, 36}	{0.95, -0.15, 0.35}	{0.0096, 0.0081, 0.0115}
8	3	{100, 94, 106}	{0.0, 0.5, -0.1}	{0.0115, 0.0115, 0.0170}
9	3	{90, 99, 111}	{0.0, 0.5, -0.1}	{0.0115, 0.0115, 0.0170}
10	3	{105, 84, 111}	{0.0, 0.5, -0.1}	{0.0115, 0.0115, 0.0170}

Table 5.1: Observation number, group membership, sizes of the blocks, means, and variances of each simulated time series.

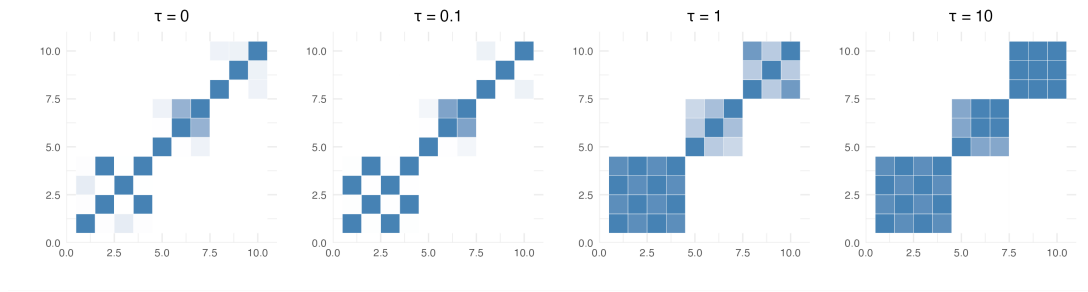
In Figure 5.3 are reported the posterior similarity matrices for each configuration averaged over the 50 replications. We notice that increasing τ the accuracy of the model increases. With $\tau = 0$, that corresponds to the non repulsive setting, we almost get 10 singletons (i.e. clusters with a single observations). Given a constant value of τ, increasing M seems to be ineffective, the posterior similarity matrices remain in fact quite the same. In Figure 5.4 are reported the number of groups in the posterior estimate for each configuration. We can see that, if we increase the repulsiveness level, the number of clusters decreases. Similarly to before, if we keep τ constant and increase M the results remain almost the same.



(a) $M = 10$

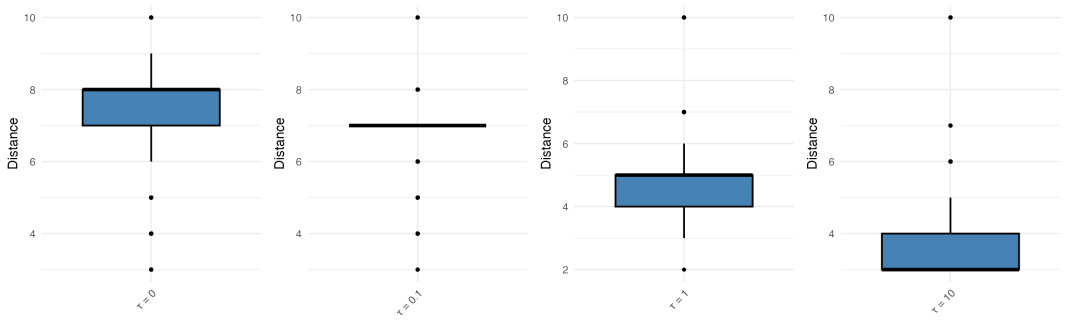


(b) $M = 50$

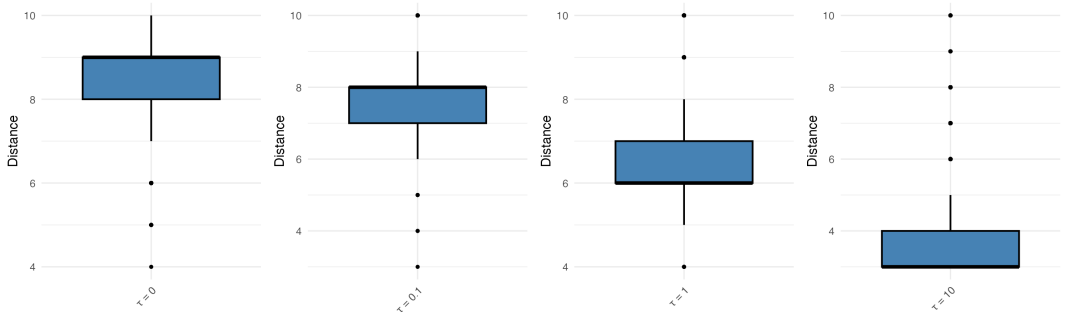


(c) $M = 100$

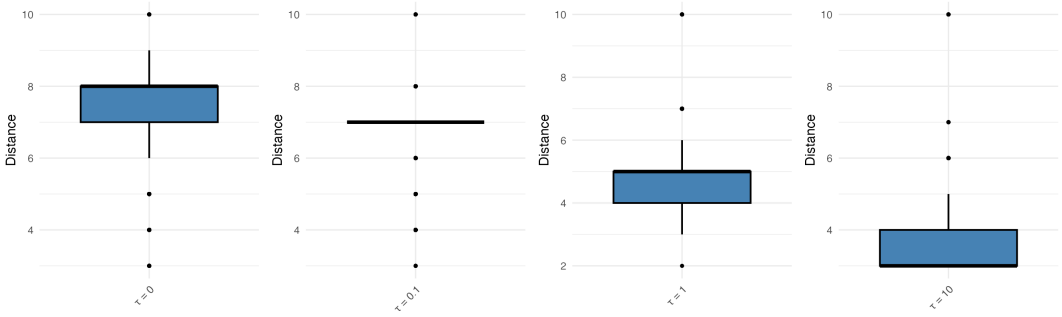
Figure 5.3: Posterior similarity matrix from simulations on synthetic data with different degrees of repulsiveness τ and number of components M in the mixture.



(a) $M = 10$



(b) $M = 50$



(c) $M = 100$

Figure 5.4: Boxplots of the number of groups in the posterior samples from simulations on synthetic data with different degrees of repulsiveness τ and different number of components M.

5.5 Real data application

We provide an illustration on financial data of this model for clustering time series with common change points. We consider the daily mean between the opening and closure stock prices of the top 50 companies for capitalization of the Standard and Poor's 500 index, from 2020-01-01 to 2022-01-02 [1]. For each observation we have a time series of 505 realisations. Data were preprocessed before running the algorithm: a linear regression was used to remove the trend from each time series to make them stationary, and the series were marginally standardized to account for the large differences in prices among some companies. To provide strong repulsiveness we set $\tau = 2$ and $\nu = 10$, moreover we set $M = 50$ to include also the possibility of having exactly one cluster for each observation. We run the algorithm for 40 000 iterations of which 20 000 are burn-in steps. The final point estimate, obtained with the SALSO procedure by Dahl et al. (2022) using the variation of information, provides 18 clusters. The composition of each cluster is reported in Table 5.2.

Cluster	Dimension	Companies
1	4	MSFT, ACN, INTU, TXN
2	4	AAPL, AMD, TMO, DHR
3	3	AMZN, JPM, ADBE
4	1	NVDA
5	4	GOOGL, GOOG, ORCL, ABT
6	3	BRK-B, CMCSA, QCOM
7	7	UNH, AVGO, PG, KO, MCD, CSCO, WFC
8	1	LLY
9	4	XOM, CVX, BAC, COP
10	4	V, MA, PM, UNP
11	5	JNJ, WMT, PFE, VZ, AMGN
12	3	HD, PEP, DIS
13	1	COST
14	1	MRK
15	1	CRM
16	1	NFLX
17	1	LIN
18	2	INTC, IBM

Table 5.2: Composition of the final clustering of the stock exchange data. For each cluster, are reported the size (number of companies) and the corresponding company codes.

We notice that 11 clusters have more than one observation, where the largest one has seven and the average dimension is 2.7. To assess the effect of repulsion, we also run the algorithm with $\nu = 0$. In this case, the model produces 27 clusters: 14 of them contained more than one company, but the average cluster size was smaller (1.8). Figure 5.5 shows the posterior similarity

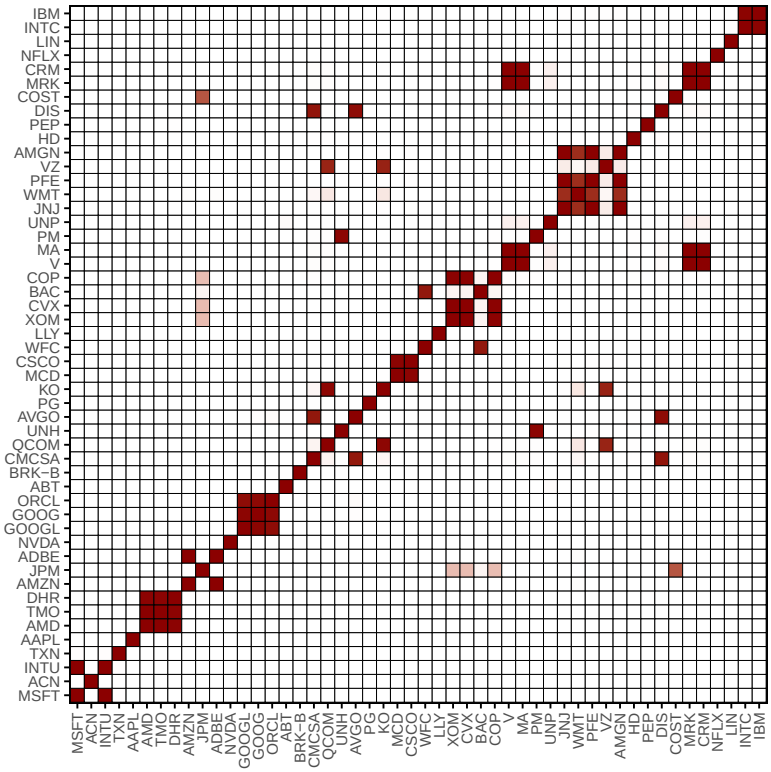
[1] Data are available from Yahoo finance https://finance.yahoo.com/

matrix under both configurations. In the non-repulsive case, the matrix is considerably sparser, indicating that companies tend to form small groups. By contrast, when repulsion is introduced ($\nu = 10$), the block structure of the matrix becomes much clearer, with larger and more cohesive clusters.

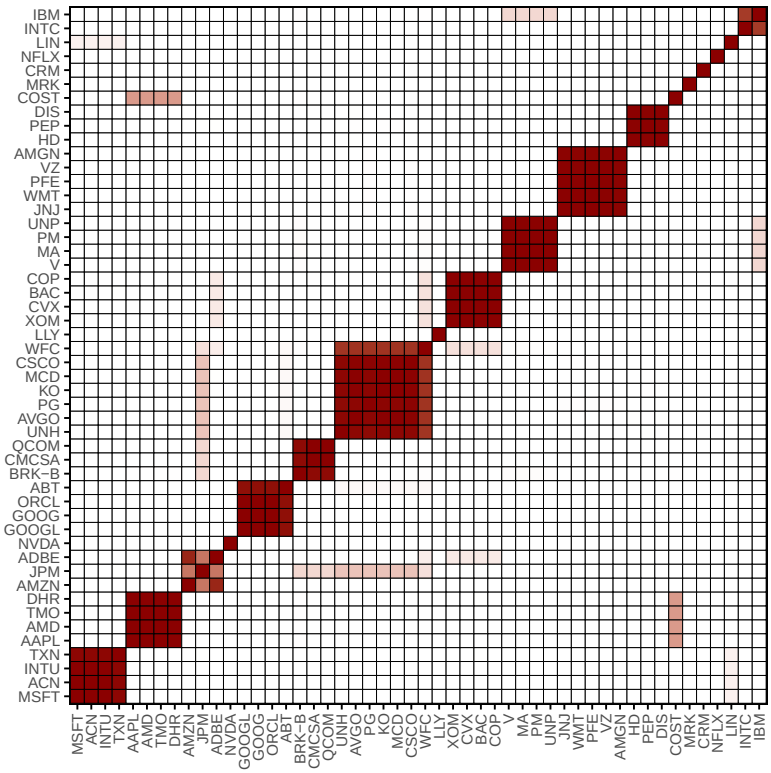
5.6 Discussion

In this chapter, we introduced a model for clustering time series that share the same number of change points, allowing these change points to occur in proximity rather than exactly at the same positions. In this sense, our work extends the one in Chapter 3, where we assume that time series within the same group must also share the exact change point locations. Compared to the model in Chapter 3, we propose a different approach based on a finite mixture model with a repulsive mechanism, which encourages clusters whose atoms are highly dissimilar. This construction promotes grouping time series with similar change points into the same clusters, as the atoms themselves are very heterogeneous.

We developed a posterior sampling algorithm inspired by Quinlan et al. (2024), which we modified to handle discrete data like ordered partitions. We investigated the properties of both the prior and the posterior distributions, the latter through an extensive simulation study. Finally, we demonstrated the method on financial stock exchange data, comparing the outcomes of the non-repulsive and repulsive settings.



(a) $\nu = 0$ (not repulsive)



(b) $\nu = 10$ (repulsive)

Figure 5.5: Comparison of posterior similarity matrix on financial data. On the top the non repulsive setting with $\nu = 0$, on the bottom the repulsive setting with $\nu = 10$.

Conclusion

In this thesis, we addressed the problem of change point detection within a Bayesian framework based on product partition models and random partitions. We proposed new methodologies as well as extensions of other well-known approaches already existing in the literature. Each contribution was studied from a theoretical point of view and motivated by applications to real data.

In Chapter 1, we provided a general introduction and an extensive literature review on change point detection, with a particular focus on Bayesian methods based on random partitions. In Chapter 2, we presented an extension of the method by Martínez and Mena (2014) for change point detection to multivariate time series with missing data, providing applications to temperature data from the City of Milan, COVID-19 data in Italy and inflation rates in the European Union. In Chapter 3 we introduced a novel methodology for clustering time series—and, more generally, time-dependent data—with common change points, illustrated through applications to Euro exchange rate data and COVID-19. All the methods introduced in Chapters 2 and 3 have been implemented in an R package, written in C++, called **BayesChange**, presented in Chapter 4 with details on the functions and examples of their implementation. Finally, in Chapter 5 we proposed a novel method for clustering time series with common but asynchronous change points, i.e., time series in the same cluster share the same change points, that do not necessarily occur at exactly the same locations but in close proximity.

Various extensions can be developed for all these projects, as well as novel methodologies inspired by them. The model for detecting change points in multivariate time series introduced in Chapter 2 can be extended to other types of data. Currently, the data are assumed to be Gaussian. However, in many applications, alternative distributional assumptions may be more appropriate. For instance, the model could be extended to count data by assuming that the realisations are distributed following a Poisson distribution. Additionally, the current model assumes a Markovian dependency structure for the time series. While this assumption simplifies inference, it may be limiting in some contexts. One could consider introducing alternative dependency structures, for example, by taking inspiration from the model of Peluso et al. (2019), which performs change point detection on vector autoregressive processes using an approach not based on random partitions. Another possible extension is to multivariate response variables whose dependency structure—represented through a directed acyclic graph (DAG; see Cowell et al. (1999) for a general introduction)—may change over time.

The strategy introduced in Chapter 3 can also be extended in several ways. In the methodology presented, we have assumed a symmetric prior distribution. However, this assumption can be relaxed by considering prior information associated with specific latent orders, as proposed

by Canale et al. (2017). Further, instead of a Dirichlet-Multinomial model, one can consider approaches based on more flexible finite-dimensional discrete prior distributions, in the spirit of Lijoi et al. (2020, 2024). However, the computational complexity of these approaches may result in models for time series clustering that are challenging to handle. Further, there is no borrowing of information across different cluster. While this assumption improves the model's tractability, having possibly dependent orders increase its flexibility. For example, recent contributions in this direction can be found in Quinlan et al. (2024). In this work we considered an approach based on the inspiring studies of Martínez and Mena (2014), where the authors restrict an EPPF from the partitions' space to the orders' space. In the same spirit, one can consider restricting a partially exchangeable partition probability function (pEPPF, see, e.g., Pitman, 1995) from the partitions' space to the orders' space, obtaining a distribution that introduces dependence among different atoms. The model discussed in this chapter can then be extended by considering such distribution to introduce dependencies among different dimensions, but preserving the same combinatorial structure and flexibility. This extension to pEPPFs can also be applied for the problem of change point detection, as an additional extension of the model in Chapter 2.

Future developments of the **BayesChange** package, presented in Chapter 4, may include the implementation of the various extensions discussed for the methods of this thesis. The most immediate extension would be the integration of the method presented in Chapter 5 into the functions for time series clustering, as well as the implementation of other approaches for estimating the final partitions—or the final ordering—and for performing uncertainty quantification, such as the WASABI method, recently introduced by Balocchi and Wade (2025).

The project in Chapter 5 is still ongoing and can be enriched by providing additional applications to other datasets, as well as deeper insights into the prior and posterior properties. As a future direction, this method could be extended by replacing the finite mixture with a fixed number of components with a mixture model with a random number of components (see Argiento and Iorio (2022) for a recent contribution on this topic). Such an extension would provide greater flexibility; however, it is not straightforward to implement, since the support we are dealing with is not continuous but discrete, consisting of partitions rather than real numbers. Another possible extension is to apply the proposed framework beyond change point detection, exploring its potential in other contexts where the goal is to cluster partitions.

Another context that can be explored is the application of Bayesian change point detection models to streaming data. In the change point literature, a distinction is usually made between offline and online methods. Offline methods detect change points on a fixed dataset, whereas online methods aim at identifying changes in real time as observations arrive sequentially; see Lai (1995) and Gillard (2015) for general works on online change point detection.

Methods for change point detection based on random partition models are typically offline. In particular for those based on Bayesian nonparametric priors, predictive rules are not available, since the observations are not globally exchangeable. Exchangeability is a necessary condition for defining the usual predictive distribution in Bayesian nonparametric models. In the change point detection context, observations are assumed to be exchangeable only within each block, while the overall dataset is not exchangeable. As a consequence, when a new observation arrives,

it cannot be assigned using a standard predictive rule. Nevertheless, a hybrid online change point detection strategy could be considered, in which the model proposed in Chapter 1 is re-run on the enlarged dataset each time a new observation becomes available. Such an approach would require an extensive investigation, in particular to assess whether a change point can be detected immediately when an observation from a new segment arrives, or whether several subsequent observations are needed before detection occurs. This represents another possible extension of this work.

Bibliography

Aghabozorgi, S., Seyed Shirkhorshidi, A., and Ying Wah, T. (2015). Time-series clustering – a decade review. *Information Systems*, 53:16–38. 41

Aitchison, J. (1982). The statistical analysis of compositional data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 44(2):139–177. 34

Anderson, D. F. and Kurtz, T. G. (2015a). *Stochastic Analysis of Biochemical Systems*. Springer International Publishing. 55

Anderson, D. F. and Kurtz, T. G. (2015b). *Stochastic Analysis of Biochemical Systems*. Springer International Publishing. 70

Andersson, H. and Britton, T. (2000). *Stochastic Epidemic Models and Their Statistical Analysis*. Springer New York. 42, 52, 54, 121

Applebaum, D. (2019). *Semigroups of Linear Operators: With Applications to Analysis, Probability and Physics*. Cambridge University Press. 120

Argiento, R. and Iorio, M. D. (2022). Is infinity that far? A Bayesian nonparametric perspective of finite mixture models. *The Annals of Statistics*, 50(5):2641 – 2663. 96

Baker, E., Barbillon, P., Fadikar, A., Gramacy, R. B., Herbei, R., Higdon, D., Huang, J., Johnson, L. R., Ma, P., Mondal, A., Pires, B., Sacks, J., and Sokolov, V. (2022). Analyzing Stochastic Computer Models: A Review with Opportunities. *Statistical Science*, 37(1):64 – 89. 42

Balocchi, C. and Wade, S. (2025). Understanding uncertainty in bayesian cluster analysis. 96

Barry, D. and Hartigan, J. A. (1992). Product Partition Models for Change Point Problems. *The Annals of Statistics*, 20(1):260 – 279. 15, 19, 24, 116

Barry, D. and Hartigan, J. A. (1993). A Bayesian Analysis for Change Point Problems. *Journal of the American Statistical Association*, 88(421):309–319. 19, 32, 60

Becker, N. G. (1993). Parametric inference for epidemic models. *Mathematical Biosciences*, 117(1):239–251. 42

Benaglia, T., Chauveau, D., Hunter, D. R., and Young, D. (2009). mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6):1–29. 60

Beraha, M., Argiento, R., Møller, J., and Guglielmi, A. (2022). Mcmc computations for bayesian mixture models using repulsive point processes. *Journal of Computational and Graphical Statistics*, 31(2):422–435. 82

Bhatt, S., Ferguson, N., Flaxman, S., Gandy, A., Mishra, S., and Scott, J. A. (2023). Semi-mechanistic bayesian modelling of covid-19 with renewal processes. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 186(4):601–615. 42

Bianchini, I., Guglielmi, A., and Quintana, F. A. (2020). Determinantal Point Process Mixtures Via Spectral Density Approach. *Bayesian Analysis*, 15(1):187 – 214. 82

Binder, D. A. (1978). Bayesian cluster analysis. *Biometrika*, 65(1):31–38. 50, 56, 63, 84

Blackwell, P. G. (2003). Bayesian inference for markov processes with diffusion and discrete components. *Biometrika*, 90(3):613–627. 27

Bong, H., Ventura, V., and Wasserman, L. (2024). Frequentist inference for semi-mechanistic epidemic models with interventions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 87(3):701–722. 42

Bouveyron, C., Côme, E., and Jacques, J. (2015). The discriminative functional mixture model for a comparative analysis of bike sharing systems. *The Annals of Applied Statistics*, 9(4):1726–1760. 50

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2016). *Time Series Analysis: Forecasting and Control*. Wiley Series in Probability and Statistics. Wiley, Hoboken, New Jersey, 5th edition. 17

Brault, V., Devijver, E., and Laclau, C. (2024). Mixture of segmentation for heterogeneous functional data. *Electronic Journal of Statistics*, 18(2):3729–3773. 42, 50

Britton, T. and Pardoux, E. (2019). *Stochastic Epidemic Models with Inference*. Springer Cham. 42

Brodsky, E. and Darkhovsky, B. S. (1993). *Nonparametric Methods in Change Point Problems*. Springer Netherlands. 17

Canale, A., Lijoi, A., Nipoti, B., and Prünster, I. (2017). On the Pitman–Yor process with spike and slab base measure. *Biometrika*, 104(3):681–697. 96

Chen, J. and Gupta, A. (2012). *Parametric Statistical Change Point Analysis*. Birkhäuser Boston. 18

Chen, J. and Gupta, A. K. (1997). Testing and locating variance changepoints with application to stock prices. *Journal of the American Statistical Association*, 92(438):739–747. 18

Chen, J., Gupta, A. K., and Gupta, A. (2000). *Parametric statistical change point analysis*, volume 192. Springer. 23

Chernoff, H. and Zacks, S. (1964). Estimating the Current Mean of a Normal Distribution which is Subjected to Changes in Time. *The Annals of Mathematical Statistics*, 35(3):999 – 1018. 18

Chib, S. (1998). Estimation and comparison of multiple change-point models. *Journal of Econometrics*, 86(2):221–241. 18

Chitwood, M. H., Russi, M., Gunasekera, K., Havumaki, J., Klaassen, F., Pitzer, V. E., Salomon, J. A., Swartwood, N. A., Warren, J. L., Weinberger, D. M., Cohen, T., and Menzies, N. A. (2022). Reconstructing the course of the covid-19 epidemic over 2020 for us states and counties: Results of a bayesian evidence synthesis model. *PLOS Computational Biology*, 18(8). 42

Choi, B. and Rempala, G. A. (2011). Inference for discretely observed stochastic kinetic networks with applications to epidemic modeling. *Biostatistics*, 13(1):153–165. 42

Corradin, R., Danese, L., KhudaBukhsh, W. R., and Ongaro, A. (2026). Model-based clustering of time-dependent observations with common structural changes. *Statistics and Computing*, 36(1):7. 21

Corradin, R., Danese, L., and Ongaro, A. (2022). Bayesian nonparametric change point detection for multivariate time series with missing observations. *International Journal of Approximate Reasoning*, 143:26–43. 21

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks*. Springer New York. 95

Cremaschi, A., Cadonna, A., Guglielmi, A., and Quintana, F. (2023). A change-point random partition model for large spatio-temporal datasets. 20

Dahl, D. B. (2006). Model-based clustering for expression data via a dirichlet process mixture model. In Do, K.-A., Müller, P., and Vannucci, M., editors, *Bayesian Inference for Gene Expression and Proteomics*, pages 201–216. Cambridge University Press. 30

Dahl, D. B., Johnson, D. J., and Müller, P. (2022). Search algorithms and loss functions for bayesian clustering. *Journal of Computational and Graphical Statistics*, 31(4):1189–1201. 37, 63, 70, 91

Di Lauro, F., KhudaBukhsh, W. R., Kiss, I. Z., Kenah, E., Jensen, M., and Rempała, G. A. (2022). Dynamic survival analysis for non-markovian epidemic models. *Journal of The Royal Society Interface*, 19(191):20220124. 121

Eddelbuettel, D. and François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18. 65

Eddelbuettel, D. and Francois, R. (2023). *RcppGSL: 'Rcpp' Integration for 'GNU GSL' Vectors and Matrices*. R package version 0.3.13. 65

Eddelbuettel, D. and Sanderson, C. (2014). Rcpparmadillo: Accelerating r with high-performance c++ linear algebra. *Computational Statistics and Data Analysis*, 71:1054–1063. 65

Engel, K.-J. and Nagel, R. (2000). *One-Parameter Semigroups for Linear Evolution Equations*. Springer-Verlag. 120

Erdman, C. and Emerson, J. W. (2007). bcp: An R package for performing a bayesian analysis of change point problems. *Journal of Statistical Software*, 23(3):1–13. 60

European Centre for Disease Prevention and Controls (2022). Database on COVID-19 daily new cases. https://www.ecdc.europa.eu/en/publications-data/data-daily-new-cases-covid-19-eueea-country. Dataset from October 2022 is no longer updated. 57

Fadikar, A., Higdon, D., Chen, J., Lewis, B., Venkatramanan, S., and Marathe, M. (2018). Calibrating a stochastic, agent-based model using quantile-based emulation. *SIAM/ASA Journal on Uncertainty Quantification*, 6(4):1685–1706. 42

Favaro, S. and Teh, Y. W. (2013). Mcmc for normalized random measure mixture models. *Statistical Science*, 28(3):335–359. 30

Fisch, A. T. M., Eckley, I. A., and Fearnhead, P. (2022). Subset multivariate collective and point anomaly detection. *Journal of Computational and Graphical Statistics*, 31(2):574–585. 81

Frühwirth-Schnatter, S. and Pamminger, C. (2010). Model-based clustering of categorical time series. *Bayesian Analysis*, 5(2):345 – 368. 41

Frühwirth-Schnatter, S. and Kaufmann, S. (2008). Model-based clustering of multiple time series. *Journal of Business & Economic Statistics*, 26(1):78–89. 41

Fuentes–García, R., Mena, R., and Walker, S. (2010). A Probability for Classification Based on the Dirichlet Process Mixture Model. *Journal of Classification*, 27:389–403. 20, 24

García, E. C. and Gutiérrez-Peña, E. (2019). Nonparametric product partition models for multiple change-points analysis. *Communications in Statistics - Simulation and Computation*, 48(7):1922–1947. 19

Ghilotti, L., Beraha, M., and Guglielmi, A. (2024). Bayesian clustering of high-dimensional data via latent repulsive mixtures. *Biometrika*, 112(2):asae059. 82

Giampino, A., Guindani, M., Nipoti, B., and Vannucci, M. (2024). Local level dynamic random partition models for changepoint detection. 20

Gikhman, I. I. and Skorokhod, A. V. (2004). *The Theory of Stochastic Processes II*. Springer Berlin Heidelberg. 120

Gillard, J. (2015). Sequential analysis: Hypothesis testing and changepoint detection. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 178(3):785–785. 96

Gleeson, J. P., Brendan Murphy, T., O'Brien, J. D., Friel, N., Bargary, N., and O'Sullivan, D. J. P. (2021). Calibrating covid-19 susceptible-exposed-infected-removed models with time-varying effective contact rates. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2214). 42

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732. 113

Green, P. J. and Richardson, S. (2001). Modelling heterogeneity with and without the dirichlet process. *Scandinavian Journal of Statistics*, 28(2):355–375. 46

Hallgren, K. L., Heard, N. A., and Turcotte, M. J. M. (2024). Changepoint Detection on a Graph of Time Series. *Bayesian Analysis*, 19(2):649 – 676. 18, 81

Hartigan, J. (1990). Partition models. *Communications in Statistics - Theory and Methods*, 19(8):2745–2756. 19

Hong, H. G. and Li, Y. (2020). Estimation of time-varying reproduction numbers underlying epidemiological processes: A new statistical tool for the covid-19 pandemic. *PLOS ONE*, 15(7). 42

Huang, J., Morsomme, R., Dunson, D., and Xu, J. (2024). Detecting changes in the transmission rate of a stochastic epidemic model. *Statistics in Medicine*, 43(10):1867–1882. 42

Inclan, C. and Tiao, G. C. (1994). Use of cumulative sums of squares for retrospective detection of changes of variance. *Journal of the American Statistical Association*, 89(427):913–923. 17

Jain, S. and Neal, R. M. (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. 46, 117, 118, 119

Jain, S. and Neal, R. M. (2007). Splitting and merging components of a nonconjugate Dirichlet process mixture model. *Bayesian Analysis*, 2(3):445 – 472. 117

James, N. A. and Matteson, D. S. (2014). ecp: An R package for nonparametric multiple change point analysis of multivariate data. *Journal of Statistical Software*, 62(7):1–25. 60

Juárez, M. A. and Steel, M. F. J. (2010). Model-based clustering of non-gaussian panel data based on skew-t distributions. *Journal of Business & Economic Statistics*, 28(1):52–66. 41

Kander, Z. and Zacks, S. (1966). Test Procedures for Possible Changes in Parameters of Statistical Distributions Occurring at Unknown Time Points. *The Annals of Mathematical Statistics*, 37(5):1196 – 1210. 18

Kermack, W. O. and McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A*, 115(772):700–721. 42

KhudaBukhsh, W. R., Bastian, C. D., Wascher, M., Klaus, C., Sahai, S. Y., Weir, M. H., Kenah, E., Root, E., Tien, J. H., and Rempała, G. A. (2023). Projecting covid-19 cases and hospital burden in ohio. *Journal of Theoretical Biology*, 561:111404. 42, 54, 56

KhudaBukhsh, W. R., Choi, B., Kenah, E., and Rempała, G. A. (2020). Survival dynamical systems: individual-level survival analysis from population-level epidemic models. *Interface Focus*, 10(1):20190048. 121

KhudaBukhsh, W. R. and Rempała, G. A. (2024). How to correctly fit an sir model to data from an seir model? *Mathematical Biosciences*, 375:109265. 54, 122

Krügel, S., Brazzale, A. R., and Kuechenhoff, H. (2017). *CPsurv: Nonparametric Change Point Estimation for Survival Data*. R package version 1.0.0. 60

Kypraios, T., Neal, P., and Prangle, D. (2017). A tutorial introduction to bayesian inference for stochastic epidemic models using approximate bayesian computation. *Mathematical Biosciences*, 287:42–53. 42

Lai, T. L. (1995). Sequential changepoint detection in quality control and dynamical systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(4):613–644. 96

Lau, J. and Green, P. (2007). Bayesian Model-Based Clustering Procedures. *Journal of Computational and Graphical Statistics*, 16(3):526–558. 50

Lavancier, F., Møller, J., and Rubak, E. (2015). Determinantal point process models and statistical inference. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 77(4):853–877. 82

Li, Y., Lund, R., and Hewaarachchi, A. (2019). Multiple changepoint detection with partial information on changepoint times. *Electronic Journal of Statistics*, 13(2):2462 – 2520. 23

Lijoi, A., Mena, R. H., and Prünster, I. (2007). Controlling the reinforcement in bayesian nonparametric mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):715–740. 39

Lijoi, A., Prünster, I., and Rigon, T. (2020). The Pitman–Yor multinomial process for mixture modelling. *Biometrika*, 107(4):891–906. 96

Lijoi, A., Prünster, I., and Rigon, T. (2024). Finite-dimensional discrete random structures and bayesian clustering. *Journal of the American Statistical Association*, 119(546):929–941. 96

Loschi, R. and Cruz, F. (2002). An analysis of the influence of some prior specifications in the identification of change points via product partition model. *Computational Statistics & Data Analysis*, 39(4):477–501. 19

Loschi, R., Cruz, F., Iglesias, P., and Arellano-Valle, R. (2003). A gibbs sampling scheme to the product partition model: an application to change-point problems. *Computers & Operations Research*, 30(3):463–482. 19

Martínez, A. F. and Mena, R. H. (2014). On a Nonparametric Change Point Detection Model in Markovian Regimes. *Bayesian Analysis*, 9(4):823 – 858. 12, 20, 21, 23, 24, 25, 30, 49, 50, 59, 61, 62, 82, 86, 95, 96, 113, 114

Matteson, D. S. and James, N. A. (2014). A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109(505):334–345. 15, 32, 116

McKee, C. and Kalli, M. (2025). Network modeling of asynchronous change-points in multivariate time series. 18, 81

Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895. 30, 32, 63, 84

Mira, A. and Petrone, S. (1997). Bayesian hierarchical nonparametric inference for change-point problems. *Bayesian Statistics*, 5. 18

Muliere, P. and Scarsini, M. (1985). Change-point problems: A bayesian nonparametric approach. *Aplikace matematiky [Applications of Mathematics]*, 30(6):397–402. 18

Neal, R. M. (2000). Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265. 30

Niels G., B. (1993). Martingale methods for the analysis of epidemic data. *Statistical Methods in Medical Research*, 2(1):93–112. 42

Niu, Y. S., Hao, N., and Zhang, H. (2016). Multiple change-point detection: A selective overview. *Statistical Science*, 31(4):611–623. 17

Okamoto, J., Stewart, N., and Li, J. (2018). *HDcpDetect: Detect Change Points in Means of High Dimensional Data*. R package version 0.1.0. 60

Paganin, S., Page, G. L., and Quintana, F. A. (2025). Informed Partition Models for Dependent Random Partitions. 20

Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1-2):100–115. 17

Page, E. S. (1957). On problems in which a change in a parameter occurs at an unknown point. *Biometrika*, 44(1-2):248–252. 17

Page, G. L., Quinlan, J. J., Curtis, S. M., and Neal, R. M. (2023). *ppmSuite: A Collection of Models that Employ Product Partition Distributions as a Prior on Partitions*. R package version 0.3.4. 60

Page, G. L., Quintana, F. A., and Dahl, D. B. (2022). Dependent modeling of temporal sequences of random partitions. *Journal of Computational and Graphical Statistics*, 31(2):614–627. 20

Pavlopoulos, V., Pham, H., Bhatt, P., Tan, Y., and Patnayakuni, R. (2024). *decp: Complete Change Point Analysis*. R package version 0.1.2. 60

Pedroso, R. C., Loschi, R. H., and Quintana, F. A. (2023). Multipartition model for multiple change point identification. *TEST*, 32(2):759–783. 19

Peluso, S., Chib, S., and Mira, A. (2019). Semiparametric Multivariate and Multiple Change-Point Modeling. *Bayesian Analysis*, 14(3):727 – 751. 18, 95

Petralia, F., Rao, V., and Dunson, D. (2012). Repulsive mixtures. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc. 82, 83, 84, 85

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, 102(2):145–158. 20, 96

Pitman, J. (2006). *Combinatorial stochastic processes*. Springer Berlin. 20, 24

Quinlan, J. J., Page, G. L., and Castro, L. M. (2024). Joint Random Partition Models for Multivariate Change Point Analysis. *Bayesian Analysis*, 19(1):21 – 48. 19, 60, 92, 96

Quinlan, J. J., Page, G. L., and Quintana, F. A. (2018). Density regression using repulsive distributions. *Journal of Statistical Computation and Simulation*, 88(15):2931–2947. 82

Quinlan, J. J., Quintana, F. A., and Page, G. L. (2021). On a class of repulsive mixture models. *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research*, 30(2):445–461. 82, 85

Quintana, F. A. and Iglesias, P. L. (2003). Bayesian clustering and product partition models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):557–574. 20

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. 59

Raúl Fierro, V. L. and Balakrishnan, N. (2015). Statistical inference on a stochastic epidemic model. *Communications in Statistics - Simulation and Computation*, 44(9):2297–2314. 42

Rempała, G. A. and KhudaBukhsh, W. R. (2023). *Dynamical Survival Analysis for Epidemic Modeling*, page 1–17. Springer International. 54

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk metropolis algorithms. *Annals of Applied Probability*, 7(1):110–120. 115

Ross, G. J. (2015). Parametric and nonparametric sequential change detection in R: The cpm package. *Journal of Statistical Software*, 66(3):1–20. 60

Same, A., Chamroukhi, F., Govaert, G., and Aknin, P. (2011). Model-based clustering and segmentation of time series with change in regime. *Advances in Data Analysis and Classification*, 5:301–321. 42

Seymour, R. G., Kypraios, T., and O'Neill, P. D. (2022). Bayesian nonparametric inference for heterogeneously mixing infectious disease models. *Proceedings of the National Academy of Sciences*, 119(10). 42

Singh, R., Ghosh, D., and Adhikari, R. (2018). Fast bayesian inference of the multivariate ornstein-uhlenbeck process. *Physical Review E*, 98:012136. 27

Smith, A. F. M. (1975). A bayesian approach to inference about a change-point in a sequence of random variables. *Biometrika*, 62(2):407–416. 18

Truong, C., Oudre, L., and Vayatis, N. (2020). Selective review of offline change point detection methods. *Signal Processing*, 167:107299. 23

Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the brownian motion. *Physical Review*, 36:823–841. 25, 48

Vannucci, M. and Do, K.-A. (2007). Bayesian model-based clustering procedures. *Journal of Computational and Graphical Statistics*, 16(3):581–601. 30

Wade, S. and Ghahramani, Z. (2018). Bayesian Cluster Analysis: Point Estimation and Credible Balls (with Discussion). *Bayesian Analysis*, 13(2):559 – 626. 50, 63

Wascher, M., Schnell, P., KhudaBukhsh, W. R., Quam, M., Tien, J., and Rempała, G. A. (2024). Estimating disease transmission in a closed population under repeated testing. *Journal of the Royal Statistical Society: Series C*. 42, 55

Xie, L., Xie, Y., and Moustakides, G. V. (2019). Asynchronous multi-sensor change-point detection for seismic tremors. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 787–791. 81

Xu, Y., Müller, P., and Telesca, D. (2016). Bayesian inference for latent biologic structure with determinantal point processes (DPP). *Biometrics*, page 955–964. 82

Yao, Y.-C. (1984). Estimation of a Noisy Discrete-Time Step Function: Bayes and Empirical Bayes Approaches. *The Annals of Statistics*, 12(4):1434 – 1447. 18, 19, 20

Young, D. S. (2013). Mixtures of regressions with changepoints. *Statistics and Computing*, 23(5):661–671. 18

Zhang, M. M., Williamson, S. A., and Pérez-Cruz, F. (2022). Accelerated parallel non-conjugate sampling for bayesian non-parametric models. *Statistics and Computing*, 32(3):1–25. 47

Appendix A

Chapter 2 supplementary materials

A.1 Proof of Proposition 1

Let $\boldsymbol{\mu} \in \mathbb{R}$ and $\Lambda \in M_{d\times d}$, with $M_{d\times d}$ denoting the space of matrices of size $d \times d$, and Λ semi-definite positive matrix. We assume a priori that $(\boldsymbol{\mu}, \Lambda) \sim \text{NIW}(\boldsymbol{m}_0, \kappa_0, \nu_0, S_0)$, where NIW denotes the Normal-Inverse-Wishart distribution, with density function

$$\pi(\boldsymbol{\mu}, \Lambda) = (2\pi)^{-\frac{d}{2}} \left|\frac{\Lambda}{\kappa_0}\right|^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{m}_0)^\intercal \left(\frac{\Lambda}{\kappa_0}\right)^{-1}(\boldsymbol{\mu}-\boldsymbol{m}_0)} \frac{|S_0|^{\frac{\nu_0}{2}}}{2^{\frac{\nu_0 d}{2}}\Gamma_d(\frac{\nu_0}{2})} |\Lambda|^{-\frac{\nu_0+d+1}{2}} \mathrm{e}^{-\frac{1}{2}\mathrm{tr}(S_0\Lambda^{-1})}$$

with $\Gamma_d(a) = \pi^{\frac{d(d-1)}{4}} \prod_{j=1}^{d} \Gamma(a + \frac{1-j}{2})$ denoting the multivariate gamma function. We further have that the distribution of a set of multivariate realisations from an isotropic and stable Ornstein–Uhlenbeck process, with equally spaced times, is given by

$$\begin{aligned}\mathcal{L}(y_{t_1}, \ldots, y_{t_{m_j}} \mid \boldsymbol{\mu}, \Lambda, \gamma) = {} & \frac{(2\pi)^{-\frac{m_j d}{2}} |\Lambda|^{-\frac{m_j}{2}}}{\prod_{i=2}^{m_j}(1-\gamma^{2\Delta_i})^{\frac{1}{2}}} \mathrm{e}^{-\frac{1}{2}(y_{t_1}-\boldsymbol{\mu})^\intercal \Lambda^{-1}(y_{t_1}-\boldsymbol{\mu})} \\ & \times \prod_{i=2}^{m_j} \mathrm{e}^{-\frac{1}{2}\left(y_{t_i}-\gamma^{\Delta_i} y_{t_{i-1}}-(1-\gamma^{\Delta_i})\boldsymbol{\mu}\right)^\intercal \frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_i})}\left(y_{t_i}-\gamma^{\Delta_i} y_{t_{i-1}}-(1-\gamma^{\Delta_i})\boldsymbol{\mu}\right)}.\end{aligned}$$

We then have that the joint distribution of $\boldsymbol{\mu}, \Lambda$ and $y_{t_1}, \ldots, y_{t_{m_j}}$, conditionally on γ, thanks to the chain rule can be written as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\mu},\Lambda, y_{t_1}, \ldots, y_{t_{m_j}} \mid \gamma) = \mathcal{L}(y_{t_1}, \ldots, y_{t_{m_j}} \mid \boldsymbol{\mu}, \Lambda, \gamma)\pi(\boldsymbol{\mu}, \Lambda) = {} & \frac{(2\pi)^{-\frac{m_j d}{2}} |\Lambda|^{-\frac{m_j}{2}}}{\prod_{i=2}^{m_j}(1-\gamma^{2\Delta_i})^{\frac{1}{2}}} \mathrm{e}^{-\frac{1}{2}(y_{t_1}-\boldsymbol{\mu})^\intercal \Lambda^{-1}(y_{t_1}-\boldsymbol{\mu})} \\ & \times \prod_{i=2}^{m_j} \mathrm{e}^{-\frac{1}{2}\left(y_{t_i}-\gamma^{\Delta_i} y_{t_{i-1}}-(1-\gamma^{\Delta_i})\boldsymbol{\mu}\right)^\intercal \frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_i})}\left(y_{t_i}-\gamma^{\Delta_i} y_{t_{i-1}}-(1-\gamma^{\Delta_i})\boldsymbol{\mu}\right)} \\ & \times (2\pi)^{-\frac{d}{2}} \left|\frac{\Lambda}{\kappa_0}\right|^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{m}_0)^\intercal \left(\frac{\Lambda}{\kappa_0}\right)^{-1}(\boldsymbol{\mu}-\boldsymbol{m}_0)} \frac{|S_0|^{\frac{\nu_0}{2}}}{2^{\frac{\nu_0 d}{2}}\Gamma_d(\frac{\nu_0}{2})} |\Lambda|^{-\frac{\nu_0+d+1}{2}} \mathrm{e}^{-\frac{1}{2}\mathrm{tr}(S_0\Lambda^{-1})}\end{aligned}$$

$$
\begin{aligned}
&= (2\pi)^{-\frac{d}{2}}(2\pi)^{-\frac{m_j d}{2}} \left|\frac{\Lambda}{\kappa_0}\right|^{-\frac{1}{2}} \frac{|S_0|^{\frac{\nu_0}{2}}}{2^{\frac{\nu_0 d}{2}}\Gamma_d(\frac{\nu_0}{2})} \frac{|\Lambda|^{-\frac{\nu_0+m_j+d+1}{2}}}{\prod_{i=2}^{m_j}(1-\gamma^{2\Delta_i})^{\frac{1}{2}}} \\
&\quad \times \exp\left\{-\frac{1}{2}\left[\boldsymbol{\mu}^\intercal\kappa_0\Lambda^{-1}\boldsymbol{\mu} - 2\boldsymbol{\mu}^\intercal\kappa_0\Lambda^{-1}\boldsymbol{m}_0 + \boldsymbol{m}_0^\intercal\kappa_0\Lambda^{-1}\boldsymbol{m}_0 + \mathrm{tr}(S_0\Lambda^{-1})\right.\right. \\
&\quad + y_{t_1}^\intercal\Lambda^{-1}y_{t_1} - 2\boldsymbol{\mu}^\intercal\Lambda^{-1}y_{t_1} + \boldsymbol{\mu}^\intercal\Lambda^{-1}\boldsymbol{\mu} + \sum_{i=2}^{m_j}\left(y_{t_i}-\gamma^{\Delta_i}y_{t_{i-1}}\right)^\intercal\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_i})}\left(y_{t_i}-\gamma^{\Delta_i}y_{t_{i-1}}\right) \\
&\quad \left.\left. -2\boldsymbol{\mu}^\intercal\Lambda^{-1}\sum_{i=2}^{m_j}\frac{(1-\gamma^{\Delta_i})}{(1-\gamma^{2\Delta_i})}(y_{t_i}-\gamma^{\Delta_i}yt_{i-1}) + \boldsymbol{\mu}^\intercal\Lambda^{-1}\boldsymbol{\mu}\sum_{i=2}^{m_j}\frac{(1-\gamma^{\Delta_i})^2}{(1-\gamma^{2\Delta_i})}\right]\right\} \\
&= (2\pi)^{-\frac{d}{2}}(2\pi)^{-\frac{m_j d}{2}} \left|\frac{\Lambda}{\kappa_0}\right|^{-\frac{1}{2}} \frac{|S_0|^{\frac{\nu_0}{2}}}{2^{\frac{\nu_0 d}{2}}\Gamma_d(\frac{\nu_0}{2})} \frac{|\Lambda|^{-\frac{\nu_0+m_j+d+1}{2}}}{\prod_{i=2}^{m_j}(1-\gamma^{2\Delta_i})^{\frac{1}{2}}} \\
&\quad \times \exp\left\{-\frac{1}{2}\left[\boldsymbol{\mu}^\intercal\Lambda^{-1}\boldsymbol{\mu}\left(\kappa_0 + \sum_{i=2}^{m_j}\frac{(1-\gamma^{\Delta_i})^2}{(1-\gamma^{2\Delta_i})} + 1\right) + \boldsymbol{m}_n^\intercal\kappa_n\Lambda^{-1}\boldsymbol{m}_n\right.\right. \\
&\quad -2\boldsymbol{\mu}^\intercal\Lambda^{-1}\frac{\kappa_n}{\kappa_n}\left(\kappa_0\boldsymbol{m}_0 + y_{t_1} + \sum_{i=2}^{m_j}\frac{(1-\gamma^{\Delta_i})}{(1-\gamma^{2\Delta_i})}\left(y_{t_i}-\gamma^{\Delta_i}y_{t_{i-1}}\right)\right) + \mathrm{tr}(S_0\Lambda^{-1}) + \mathrm{tr}(y_{t_1}y_{t_1}^\intercal\Lambda^{-1}) \\
&\quad \left.\left. +\mathrm{tr}\left(\sum_{i=2}^{m_j}\frac{(y_{t_i}-\gamma^{\Delta_i}y_{t_{i-1}})(y_{t_i}-\gamma^{\Delta_i}y_{t_{i-1}})^\intercal}{(1-\gamma^{2\Delta_i})}\Lambda^{-1}\right) + \mathrm{tr}(\kappa_0\boldsymbol{m}_0\boldsymbol{m}_0^\intercal\Lambda^{-1}) - \mathrm{tr}(\kappa_n\boldsymbol{m}_n\boldsymbol{m}_n^\intercal\Lambda^{-1})\right]\right\} \\
&= (2\pi)^{-\frac{d}{2}}\left|\frac{\Lambda}{\kappa_0}\right|^{-\frac{1}{2}} \frac{|\Lambda|^{-\frac{\nu_n+d+1}{2}}}{\prod_{i=2}^{m_j}(1-\gamma^{2\Delta_i})^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{m}_n)^\intercal\left(\frac{\Lambda}{\kappa_n}\right)^{-1}(\boldsymbol{\mu}-\boldsymbol{m}_n)\right\} \\
&\quad \times (2\pi)^{-\frac{m_j d}{2}} \frac{|S_0|^{\frac{\nu_0}{2}}}{2^{\frac{\nu_0 d}{2}}\Gamma_d(\frac{\nu_0}{2})} \exp\left\{-\frac{1}{2}\mathrm{tr}(S_n\Lambda^{-1})\right\}
\end{aligned}
$$

which identifies the kernel of a NIW$(\kappa_n, \boldsymbol{m}_n, \nu_n, S_n)$ distribution, with parameters equal to

$$
\begin{aligned}
\kappa_n &= \kappa_0 + \sum_{i=2}^{m_j}\frac{(1-\gamma^{\Delta_i})^2}{(1-\gamma^{2\Delta_i})} + 1 \\
m_n &= \frac{1}{\kappa_n}\left[\kappa_0\boldsymbol{m}_0 + y_{t_1} + \sum_{i=2}^{m_j}\frac{(1-\gamma^{\Delta_i})}{(1-\gamma^{2\Delta_i})}\left(y_{t_i}-\gamma^{\Delta_i}y_{t_{i-1}}\right)\right] \\
\nu_n &= \nu_0 + m_j \\
S_n &= S_0 + y_{t_1}y_{t_1}^\intercal + \sum_{i=2}^{m_j}\frac{(y_{t_i}-\gamma^{\Delta_i}y_{t_{i-1}})(y_{t_i}-\gamma^{\Delta_i}y_{t_{i-1}})^\intercal}{(1-\gamma^{2\Delta_i})} + \kappa_0\boldsymbol{m}_0\boldsymbol{m}_0^\intercal - \kappa_n\boldsymbol{m}_n\boldsymbol{m}_n^\intercal
\end{aligned}
$$

A.2 Proof of Proposition 2

Let $\boldsymbol{\mu} \in \mathbb{R}$ and $\Lambda \in M_{d\times d}$, with $M_{d\times d}$ denoting the space of matrices of size $d \times d$, and Λ positive semi-definite. We assume a priori that $(\boldsymbol{\mu}, \Lambda) \sim \mathrm{NIW}(\boldsymbol{m}_0, \kappa_0, \nu_0, S_0)$, where NIW denotes the Normal-Inverse-Wishart distribution. We can integrate $\mathcal{L}(\boldsymbol{\mu}, \Lambda, y_{t_1}, \ldots, y_{t_{m_j}} \mid \gamma)$, described in

A.1, with respect to $\boldsymbol{\mu}$ and Λ, obtaining

$$\begin{aligned}
\mathcal{L}(y_{t_1},\dots,y_{t_{m_j}} \mid \gamma) &= \int_{\mathbb{R}^d}\int_{\mathbb{M}_{d\times d}} \mathcal{L}(\boldsymbol{\mu},\Lambda,y_{t_1},\dots,y_{t_{m_j}})\mathrm{d}\boldsymbol{\mu}\mathrm{d}\Lambda \\
&= \int_{\mathbb{R}^d}\int_{\mathbb{M}_{d\times d}} \frac{(2\pi)^{-\frac{d}{2}}}{\kappa_0^{-\frac{d}{2}}}|\Lambda|^{-\frac{1}{2}}\mathrm{e}^{-\frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{m}_n)^\intercal\left(\frac{\Lambda}{\kappa_n}\right)^{-1}(\boldsymbol{\mu}-\boldsymbol{m}_n)} \\
&\times \frac{(\pi)^{-\frac{m_j d}{2}}}{\prod_{i=2}^{m_j}(1-\gamma^{2\Delta_i})^{\frac{1}{2}}}\frac{|S_0|^{\frac{\nu_0}{2}}}{\Gamma_d(\frac{\nu_0}{2})}\frac{|\Lambda|^{-\frac{\nu_0+m_j+d+1}{2}}}{2^{\frac{(\nu_0+m_j)d}{2}}}\mathrm{e}^{-\frac{1}{2}\mathrm{tr}(S_n\Lambda^{-1})}\mathrm{d}\boldsymbol{\mu}\mathrm{d}\Lambda \\
&= \frac{\kappa_n^{-\frac{d}{2}}}{\kappa_0^{-\frac{d}{2}}}\frac{\pi^{-\frac{m_j d}{2}}}{\prod_{i=2}^{m_j}(1-\gamma^{2\Delta_i})^{\frac{1}{2}}}\frac{|S_0|^{\frac{\nu_0}{2}}}{\Gamma_d(\frac{\nu_0}{2})}\frac{\Gamma_d(\frac{\nu_n}{2})}{|S_n|^{\frac{\nu_n}{2}}}
\end{aligned}$$

A.3 Proof of Proposition 3

Let $\boldsymbol{\mu}\in\mathbb{R}$ and $\Lambda\in M_{d\times d}$, with $M_{d\times d}$ denoting the space of positive semi-definite matrices of size $d\times d$. Let $y_{t_1},\dots,y_{t_{m_j}}$ a set of realisations from the Orstein-Uhlenbeck process, taking values on $\mathbb{R}^d$. Let $y_{(t_j)} = (y_{t_1},\dots,y_{t_{m_j}})\setminus y_{t_j}$. Then we have three distinct case to impute the missing data.

- $j=1$, then we have

$$\begin{aligned}
\mathcal{L}(y_{t_1}\mid y_{(t_1)},\gamma) &\propto \exp\left\{-\frac{1}{2}(y_{t_1}-\boldsymbol{\mu})^\intercal\Lambda^{-1}(y_{t_1}-\boldsymbol{\mu})\right. \\
&\left.-\frac{1}{2}\left(y_{t_2}-\gamma^{\Delta_2}y_{t_1}-(1-\gamma^{\Delta_2})\boldsymbol{\mu}\right)^\intercal\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_2})}\left(y_{t_2}-\gamma^{\Delta_2}y_{t_1}-(1-\gamma^{\Delta_2})\boldsymbol{\mu}\right)\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[y_{t_1}^\intercal\Lambda^{-1}y_{t_1}-2y_{t_1}^\intercal\Lambda^{-1}\boldsymbol{\mu}+y_{t_1}^\intercal\frac{\gamma^{2\Delta_2}\Lambda^{-1}}{(1-\gamma^{2\Delta_2})}y_{t_1}-2y_{t_1}^\intercal\frac{\gamma^{\Delta_2}\Lambda^{-1}}{(1-\gamma^{2\Delta_2})}(y_{t_2}-(1-\gamma^{\Delta_2})\boldsymbol{\mu})\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\left[y_{t_1}^\intercal\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_2})}y_{t_1}-2y_{t_1}^\intercal\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_2})}\left(\gamma^{\Delta_2}y_{t_2}+(1-\gamma^{\Delta_2})\boldsymbol{\mu}\right)\right.\right. \\
&\left.\left.\pm\left(\gamma^{\Delta_2}y_{t_2}+(1-\gamma^{\Delta_2})\boldsymbol{\mu}\right)^\intercal\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_2})}\left(\gamma^{\Delta_2}y_{t_2}+(1-\gamma^{\Delta_2})\boldsymbol{\mu}\right)\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}(y_{t_1}-\boldsymbol{\lambda})\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_2})}(y_{t_1}-\boldsymbol{\lambda})\right\}
\end{aligned}$$

  which identifies the kernel of a multivariate Gaussian distribution with covariance parameter equal to $\Lambda(1-\gamma^{2\Delta_2})$ and expected value $\boldsymbol{\lambda}=\gamma^{\Delta_2}y_{t_2}+(1-\gamma^{\Delta_2})\boldsymbol{\mu}$.

- $1<j<T$, then we have

$$\begin{aligned}
\mathcal{L}(y_{t_j}\mid y_{(t_j)},\gamma) &\propto \exp\left\{-\frac{1}{2}\left(y_{t_j}-\gamma^{\Delta_j}y_{t_{j-1}}-(1-\gamma^{\Delta_j})\boldsymbol{\mu}\right)^\intercal\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_j})}\left(y_{t_j}-\gamma^{\Delta_j}y_{t_{j-1}}-(1-\gamma^{\Delta_j})\boldsymbol{\mu}\right)\right. \\
&\left.-\frac{1}{2}\left(y_{t_{j+1}}-\gamma^{\Delta_{j+1}}y_{t_j}-(1-\gamma^{\Delta_{j+1}})\boldsymbol{\mu}\right)^\intercal\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_{j+1}})}\left(y_{t_{j+1}}-\gamma^{\Delta_{j+1}}y_{t_j}-(1-\gamma^{\Delta_{j+1}})\boldsymbol{\mu}\right)\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[y_{t_j}^\intercal\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_j})}y_{t_j}-2y_{t_j}\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_j})}\left(\gamma^{\Delta_j}y_{t_{j-1}}+(1-\gamma^{\Delta_j})\boldsymbol{\mu}\right)\right.\right. \\
&\left.\left.+\gamma^{2\Delta_{j+1}}y_{t_j}^\intercal\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_{j+1}})}y_{t_j}-2\gamma^{\Delta_{j+1}}y_{t_j}^\intercal\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_{j+1}})}(y_{t_{j+1}}-\left(1-\gamma^{\Delta_{j+1}}\right)\boldsymbol{\mu})\right]\right\}
\end{aligned}$$

$$
\begin{aligned}
&= \exp\left\{-\frac{1}{2}\left[y_{t_j}^\intercal \Lambda^{-1}\left(\frac{1}{(1-\gamma^{2\Delta_j})}+\frac{\gamma^{2\Delta_{j+1}}}{(1-\gamma^{2\Delta_{j+1}})}\right)y_{t_j}\right.\right.\\
&-2y_{t_j}^\intercal\Lambda^{-1}\left(\frac{\gamma^{\Delta_j}(1-\gamma^{2\Delta_{j+1}})y_{t_{j-1}}+\gamma^{\Delta_{j+1}}(1-\gamma^{2\Delta_j})y_{t_{j+1}}}{(1-\gamma^{2\Delta_j})(1-\gamma^{2\Delta_{j+1}})}\right.\\
&\left.\left.+\ \frac{(1-\gamma^{\Delta_j})(1-\gamma^{2\Delta_{j+1}})\boldsymbol{\mu}-\gamma^{\Delta_{j+1}}(1-\gamma^{\Delta_{j+1}})(1-\gamma^{2\Delta_j})\boldsymbol{\mu}}{(1-\gamma^{2\Delta_j})(1-\gamma^{2\Delta_{j+1}})}\right)\right]\right\}\\
&\propto \exp\left\{-\frac{1}{2}(y_{t_j}-\boldsymbol{\lambda})\Lambda^{-1}\left(\frac{1-\gamma^{2(\Delta_j+\Delta_{j+1})}}{(1-\gamma^{2\Delta_j})(1-\gamma^{2\Delta_{j+1}})}\right)(y_{t_j}-\boldsymbol{\lambda})\right\}
\end{aligned}
$$

which identifies the kernel of a multivariate Gaussian distribution with covariance matrix equal to $\Lambda\left(\frac{(1-\gamma^{2\Delta_j})(1-\gamma^{2\Delta_{j+1}})}{1-\gamma^{2(\Delta_j+\Delta_{j+1})}}\right)$ and expected value

$$
\begin{aligned}
\boldsymbol{\lambda} = &\frac{\gamma^{\Delta_j}(1-\gamma^{2\Delta_{j+1}})y_{t_{j-1}}+\gamma^{\Delta_{j+1}}(1-\gamma^{2\Delta_j})y_{t_{j+1}}}{1-\gamma^{2(\Delta_j+\Delta_{j+1})}}\\
&+\frac{(1-\gamma^{\Delta_j})(1-\gamma^{2\Delta_{j+1}})\boldsymbol{\mu}-\gamma^{\Delta_{j+1}}(1-\gamma^{\Delta_{j+1}})(1-\gamma^{2\Delta_j})\boldsymbol{\mu}}{1-\gamma^{2(\Delta_j+\Delta_{j+1})}}.
\end{aligned}
$$

- $j = m_j$, then we have

$$
\begin{aligned}
\mathcal{L}(y_{t_j} \mid y_{(t_j)},\gamma) \propto \exp\Big\{ &-\frac{1}{2}\left(y_{t_{m_j}}-\gamma^{\Delta_{m_j}}y_{t_{n_j-1}}-(1-\gamma^{\Delta_{m_j}})\boldsymbol{\mu}\right)^\intercal\\
&\frac{\Lambda^{-1}}{(1-\gamma^{2\Delta_{m_j}})}\left(y_{t_{m_j}}-\gamma^{\Delta_{m_j}}y_{t_{m_j-1}}-(1-\gamma^{\Delta_{m_j}})\boldsymbol{\mu}\right)\Big\}
\end{aligned}
$$

which identifies the kernel of a multivariate Gaussian distribution with covariance parameter equal to $\Lambda(1-\gamma^{2\Delta_{m_j}})$ and expected value $\boldsymbol{\lambda} = \gamma^{\Delta_{m_j}}y_{t_{n_j-1}} + (1-\gamma^{\Delta_{m_j}})\boldsymbol{\mu}$.

A.4 Proof of Proposition 4

Hereby we consider two distinct observations from an isotropic and stable OU$(\boldsymbol{\mu},\Lambda,\gamma)$ process, with time increments Δ_i. Then we have

$$
y_{t_i} \mid y_{t_{i-1}} \sim \mathrm{N}(\gamma^{\Delta_i}y_{t_{i-1}} + (1-\gamma^{\Delta_i})\boldsymbol{\mu}, \Lambda(1-\gamma^{2\Delta_i})).
$$

We then have

$$
\begin{aligned}
\mathrm{Cov}(y_{t_i}, y_{t_{i-1}}) &= \mathbb{E}[y_{t_i}y_{t_{i-1}}^\intercal] - \mathbb{E}[y_{t_i}]\mathbb{E}[y_{t_{i-1}}^\intercal]\\
&= \mathbb{E}[\mathbb{E}[y_{t_i}y_{t_{i-1}}^\intercal \mid y_{t_{i-1}}]] - \mathbb{E}[y_{t_i}]\mathbb{E}[y_{t_{i-1}}^\intercal]\\
&= \gamma^{\Delta_i}\mathbb{E}[y_{t_{i-1}}y_{t_{i-1}}^\intercal] \pm \mathbb{E}[y_{t_{i-1}}]\mathbb{E}[y_{t_{i-1}}^\intercal] + (1-\gamma^{\Delta_i})\boldsymbol{\mu}\mathbb{E}[y_{t_{i-1}}] - \mathbb{E}[y_{t_i}]\mathbb{E}[y_{t_{i-1}}^\intercal]\\
&= \gamma^{\Delta_i}\mathbb{E}[y_{t_{i-1}}y_{t_{i-1}}^\intercal] - \mathbb{E}[y_{t_{i-1}}]\mathbb{E}[y_{t_{i-1}}^\intercal]\\
&= \gamma^{\Delta_i}\mathrm{Var}(y_{t_{i-1}})
\end{aligned}
$$

which concludes the proof of Proposition 4.

A.5 RJMCMC algorithm with missing data

We briefly describe the algorithm to obtain realisations of the posterior distribution of the model introduced in section 2.2, which is a reversible-jump MCMC (Green, 1995). The algorithm reported in the following pseudocode is fully inspired by the sampling strategy introduced in Martínez and Mena (2014), but extended to the multivariate case with possibly missing observations.

We denote by $\mathcal{L}(y_1, \dots, y_n \mid \rho_n, \gamma)$ the distribution of $y_1, \dots, y_n$ conditionally on a specific ordering described by its composition ρ_n. In force of Equation 2.12, we have

$$\begin{aligned}\mathcal{L}(y_{1:n} \mid \rho_n, \gamma) &= \prod_{j=1}^{k} \mathcal{L}(y_{m_j+1}, \dots, y_{m_j+m_j} \mid \gamma) \\ &= \frac{\kappa_0^{kd/2} |S_0|^{k\nu_0/2}}{\pi^{nd/2}(1-\gamma^2)^{(n-k)/2} \left(\Gamma_d(\frac{\nu_0}{2})\right)^k} \prod_{j=1}^{k} \frac{\Gamma_d(\frac{\nu_{m_j}}{2})}{\kappa_{m_j}^{d/2} |S_{m_j}|^{\nu m_j/2}}\end{aligned} \tag{A.1}$$

and we further denote by $\mathcal{L}(y_{1:n}, \rho_n \mid \gamma)$ the product of the last term in Equation A.1 with the form of the restricted EPPF in Equation 2.3 evaluated on the composition ρ_n, i.e.

$$\mathcal{L}(y_{1:n}, \rho_n \mid \gamma) = \frac{n! \kappa_0^{kd/2} |S_0|^{k\nu_0/2} \prod_{j=1}^{k-1}(\vartheta + j\sigma)}{k!(\vartheta+1)_{n-1} \pi^{nd/2}(1-\gamma^2)^{(n-k)/2} \left(\Gamma_d(\frac{\nu_0}{2})\right)^k} \prod_{j=1}^{k} \frac{\Gamma_d(\frac{\nu_{m_j}}{2})(1-\sigma)_{m_j-1}}{m_j! \kappa_{m_j}^{d/2} |S_{m_j}|^{\nu m_j/2}}.$$

We can produce a sample from the posterior distribution of $\rho_n \mid y_1, \dots, y_n$, with possibly missing observations, as follow.

Sampling scheme for HPY mixture model.

Initialize the composition latent in the data $\rho_n = (m_1, \dots, m_k)$ with k distinct groups;

initialize σ, ϑ, γ and all the missing observations;
for $r = 1, \dots, R$:

(i) perform a split and merge step to update Ψ_n:

(a) with probability $q\mathbb{1}_{[1<k<T]} + \mathbb{1}_{[k=n]}$ perform a split:

- choose j uniformly from $\{j : 1 \le j \le k, m_j > 1\}$;
- choose ℓ uniformly from $\{1, \dots, m_j - 1\}$, such that the proposed random order is defined via the composition

$$\rho_n^* = (n_1, \dots, m_{j-1}, \ell, m_j - \ell, \dots, m_k);$$

- accept the new composition ρ_n^* with probability

$$\alpha = \min\left\{1, \left(\frac{1-q}{q}\right) \frac{\mathcal{L}(y_{1:n}, \rho_n^* \mid \gamma)}{\mathcal{L}(y_{1:n}, \rho_n \mid \gamma)} \frac{n_{g,1}(n_\ell - 1)}{k}\right\} \qquad \text{if } 1 \le k < T,$$

$$\alpha = \min\left\{1, (1-q)(n-1)\frac{\mathcal{L}(y_{1:n}, \rho_n^* \mid \gamma)}{\mathcal{L}(y_{1:n}, \rho_n \mid \gamma)}\right\} \qquad \text{if } k = T,$$

with $n_{g,1} = \#\{m_\ell \in \rho_n : m_\ell > 1\}$, otherwise stay on the previous composition ρ_n;

(b) otherwise, with probability $(1-q)\mathbb{1}_{[1<k<T]}$, perform a merge:

- choose j uniformly from $\{1, \ldots, k-1\}$;
- set the proposed random order as

$$\rho_n^* = (n_1, \ldots, m_j + m_{j+1}, \ldots, m_k);$$

- accept the new composition ρ_n^* with probability

$$\alpha = \min\left\{1, \left(\frac{q}{1-q}\right)\frac{\mathcal{L}(y_{1:n}, \rho_n^* \mid \gamma)}{\mathcal{L}(y_{1:n}, \rho_n \mid \gamma)}\frac{k-1}{n_{g,1}^*(m_j + m_{j+1} - 1)}\right\} \qquad \text{if } 1 < k < T,$$

$$\alpha = \min\left\{1, q(n-1)\frac{\mathcal{L}(y_{1:n}, \rho_n^* \mid \gamma)}{\mathcal{L}(y_{1:n}, \rho_n \mid \gamma)}\right\} \qquad \text{if } k = T,$$

$n_{g,1}^* = \#\{n_\ell \in \rho_n^* : m_\ell > 1\}$, otherwise stay on the previous composition ρ_n;

(ii) if $k > 1$ perform a reshuffle step:

(a) choose j uniformly from $\{1, \ldots, k-1\}$;

(b) choose ℓ uniformly from $\{1, \ldots, m_j + m_{j+1} - 1\}$ and set the proposed random order as

$$\rho_n^* = (n_1, \ldots, m_{j-1}, \ell, m_j + m_{j+1} - \ell, \ldots, m_k);$$

(c) accept the new composition ρ_n^* with probability

$$\alpha = \min\left\{1, \frac{\mathcal{L}(y_{1:n}, \rho_n^* \mid \gamma)}{\mathcal{L}(y_{1:n}, \rho_n \mid \gamma)}\right\},$$

otherwise stay on the previous composition ρ_n;

(iii) update σ and ϑ from their posterior distribution, according to Martínez and Mena (2014);

(iv) update γ from its full conditional distribution performing a Metropolis-Hastings step on a transformed scale with

$$\tau = g(\gamma) = \log\left(\frac{\gamma}{1-\gamma}\right), \qquad \gamma = g^{-1}(\tau) = \frac{\mathrm{e}^\tau}{1+\mathrm{e}^\tau}$$

with Gaussian proposal $\tau^* = \tau + T$, $T \sim N(0, \sigma_T^2)$, and acceptance rate

$$\alpha = \min\left\{1, \frac{\mathcal{L}(y_{1:n}, \rho_n \mid g^{-1}(\tau^*))\pi(g^{-1}(\tau^*))|\frac{\mathrm{d}}{\mathrm{d}\tau}g^{-1}(\tau^*)|}{\mathcal{L}(y_{1:n}, \rho_n \mid g^{-1}(\tau))\pi(g^{-1}(\tau))|\frac{\mathrm{d}}{\mathrm{d}\tau}g^{-1}(\tau)|}\right\}$$

where the variance of the proposal distribution σ_T^2 can be tuned to obtain an optimal acceptance rate (see Roberts et al., 1997), and $\pi(\cdot)$ denotes the prior distribution on γ; _

(v) perform an imputation step for the missing values: for $i = 1, \ldots, T$:

(a) if $Z_i = 0$:

– let j denotes the group label of the i-th observation; sample $\boldsymbol{\mu}_j$ and Λ_j from their posterior distribution, according to Equation 2.11, such that

$$\boldsymbol{\mu}_j, \Lambda_j \mid y_{\widetilde{m}_j+1}, \ldots, y_{\widetilde{m}_j+m_j}, \gamma \sim \mathrm{NIW}(\boldsymbol{m}_{m_j}, \kappa_{m_j}, \nu_{m_j}, S_{m_j})$$

with $\widetilde{m}_j = \sum_{\ell=1}^{j-1} m_\ell$, and

$$\begin{aligned}
\kappa_{m_j} &= \kappa_0 + (m_j - 1)\frac{(1-\gamma)^2}{(1-\gamma^2)} + 1, \\
m_{m_j} &= \frac{1}{\kappa_{m_j}}\left[\kappa_0 \boldsymbol{m}_0 + y_{\widetilde{m}_j+1} + \frac{(1-\gamma)}{(1-\gamma^2)}\sum_{i=\widetilde{m}_j+2}^{\widetilde{m}_j+m_j}(y_i - \gamma y_{i-1})\right], \\
\nu_{m_j} &= \nu_0 + m_j, \\
S_{m_j} &= S_0 + y_{m_j+1}y_{m_j+1}^\intercal + \frac{1}{(1-\gamma^2)}\sum_{i=\widetilde{m}_j+2}^{\widetilde{m}_j+m_j}(y_i - \gamma y_{i-1})(y_i - \gamma y_{i-1})^\intercal \\
&\quad + \kappa_0 \boldsymbol{m}_0 \boldsymbol{m}_0^\intercal - \kappa_{m_j}\boldsymbol{m}_{m_j}\boldsymbol{m}_{m_j}^\intercal;
\end{aligned}$$

– perform missing values imputation for the i-th observation, according to

$$y_i \mid Z_i, y_{i-1}, y_{i+1}, \boldsymbol{\mu}_j^*, \Lambda_j^*, \gamma \sim N(\boldsymbol{\lambda}, \Phi)$$

with

$$\begin{aligned}
&\boldsymbol{\lambda} = \gamma y_{i+1} + (1-\gamma)\boldsymbol{\mu}_j, \quad \Phi = \Lambda_j(1-\gamma^2), && \text{if } i = \widetilde{m}_j + 1, \\
&\boldsymbol{\lambda} = \frac{\gamma(y_{i+1} + y_{i-1})}{(1+\gamma^2)} + \frac{(1-\gamma)^2\boldsymbol{\mu}_j}{(1+\gamma^2)}, \ \Phi = \Lambda_j\left(\frac{1-\gamma^2}{1+\gamma^2}\right), && \text{if } \widetilde{m}_j + 1 < i < \widetilde{m}_j + m_j, \\
&\boldsymbol{\lambda} = \gamma y_{i-1} + (1-\gamma)\boldsymbol{\mu}_j, \quad \Phi = \Lambda_j(1-\gamma^2), && \text{if } i = \widetilde{m}_j + m_j,
\end{aligned}$$

with $\widetilde{m}_j = \sum_{\ell=1}^{j-1} m_\ell$.

A.6 Comparison

Scenario	Model (12)		C1		C2	
	$\hat{k}$	$\text{VI}(\hat{\rho}, \rho_0)$	$\hat{k}$	$\text{VI}(\hat{\rho}, \rho_0)$	$\hat{k}$	$\text{VI}(\hat{\rho}, \rho_0)$
1	3	0	3.00	0.081	4.02	0.036
3	3.12	0.001	2.36	0.068	11.06	0.150
4	5.14	0.038	2.22	0.085	14.26	0.173

Table A.1: Comparisons of the proposed model (Equation (2.13) and two competitors: C1 (Matteson and James, 2014) and C2 (Barry and Hartigan, 1992). For each model, we report the number of blocks in the point estimate of the latent order $\hat{k}$ and the variation of information distance $\text{VI}(\hat{\rho}, \rho_0)$, averaged over 50 replications.

Appendix B

Chapter 3 supplementary materials

B.1 Additional details on the split and merge algorithm

In this section we provide further details on the algorithm presented in Section 3.4, which can be helpful for future implementations of the model discussed in the manuscript. The split and merge strategy we used in the manuscript is based on the preliminary studies of Jain and Neal (2004, 2007), where we make use of some results of the latter for non-conjugate distributions. Let us first recall the general specification of the model, with

$$\begin{aligned}
\boldsymbol{y}_i \mid \rho_i, \boldsymbol{\theta}_i^* &\sim \prod_{j=1}^{m_i} \prod_{t=t_{i,j}^-}^{t_{i,j}^+} \mathcal{L}(y_{i,t} \mid y_{i,t-1}, \theta_{i,j}^*), \quad i = 1, \dots, n, \\
\rho_i \mid \widetilde{p}(\rho) &\overset{\text{iid}}{\sim} \widetilde{p}(\rho), \quad i = 1, \dots, n, \\
(\pi_1, \dots, \pi_{2^{T-1}}) &\sim \text{DIR}(\alpha_1, \dots, \alpha_{2^{T-1}}), \\
\theta_{i,j}^* &\overset{\text{iid}}{\sim} P_0(\theta), \quad j = 1, \dots, m_i, \ i = 1, \dots, n,
\end{aligned} \tag{B.1}$$

whereas

$$\widetilde{p}(\rho) = \sum_{r=1}^{2^{T-1}} \pi_r \delta_{\widetilde{\rho}_r}(\rho), \tag{B.2}$$

$\{\widetilde{\rho}_1, \dots, \widetilde{\rho}_{2^{T-1}}\}$ denote all the possible orders of T elements, and $(\pi_1, \dots, \pi_{2^{T-1}})$ is a vector of probabilities taking values in the $(2^{T-1} - 1)$-dimensional simplex space.

Before discussing the sampling step, we introduce some quantities which will be needed in the derivation of the acceptance rates. We first note that the distribution of a sequence of elements $\mathcal{R} = \{\rho_1, \dots, \rho_n\}$ can be decomposed into the distribution of the partition structure λ times the distribution of the unique elements $\mathcal{R}^* = \{\rho_1^*, \dots, \rho_k^*\}$, with $k \leq n$, and

$$\mathcal{L}(\mathcal{R}) = \mathcal{L}(\lambda)\mathcal{L}(\mathcal{R}^*).$$

The first term is given by the EPPF of a Dirichlet-categorical model, with

$$\mathcal{L}(\lambda) = \frac{\Gamma(\alpha^+)}{\Gamma(\alpha^+ + n)} \prod_{r=1}^{k} \frac{\Gamma(\alpha_r + n_r)}{\Gamma(\alpha_r)},$$

with $\lambda = \{B_1, \ldots, B_k\}$ and $n_j = |B_j|$. The second term is the probability associated with a specific set of unique values. As far as all the values have the same prior guess, the joint distribution of a specific subset is given by the law of a uniform sampling without replacement. Hence, we have

$$\mathcal{L}(\mathcal{R}^*) = 1 - \prod_{j=0}^{k-1} \left(1 - \frac{1}{2^{T-1} - j}\right),$$

where k is the number of groups. Finally, we denote by

$$q(\lambda^{(N)}, \mathcal{R}^{(N)} \mid \lambda^{(O)}, \mathcal{R}^{(O)}, \mathcal{Y})$$

the proposal from a previous state $(\lambda^{(O)}, \mathcal{R}^{(O)})$ to a new state $(\lambda^{(N)}, \mathcal{R}^{(N)})$, where we emphasize the dependence on $\mathcal{Y}$ of the mixture strategy as proposal distribution for the new orders. Further, we have that the proposal factorises into two terms

$$q(\lambda^{(N)}, \mathcal{R}^{(N)} \mid \lambda^{(O)}, \mathcal{R}^{(O)}, \mathcal{Y}) = q(\lambda^{(N)} \mid \lambda^{(O)}) q(\mathcal{R}^{(N)} \mid \mathcal{Y}), \tag{B.3}$$

whereas the first term in the left side of the previous equation is described in Jain and Neal (2004) and does not depend on $\mathcal{Y}$, while the second term is a product of independent proposals following the mixture of posteriors discussed in the manuscript, i.e.

$$\psi(\rho \mid \mathcal{Y}) = \sum_{i=1}^{n} \frac{1}{n} \mathcal{L}(\rho \mid \boldsymbol{y}_i). \tag{B.4}$$

We can now describe a detailed single step of the split and merge strategy for clustering multiple time series. Suppose we are performing the generic m step of the algorithm to cluster the observed data $\mathcal{Y}$, assuming we are on a current state of the chain, with $\lambda^{(m-1)} = \{B_1^{(m-1)}, \ldots, B_k^{(m-1)}\}$ being the latent partition of the data in k groups at the current state and $\mathcal{R}^{*(m-1)} = \{\rho_1^{*(m-1)}, \ldots, \rho_k^{*(m-1)}\}$ denotes the unique values of the latent orders out of $\mathcal{R}^{(m-1)} = \{\rho_1^{(m-1)}, \ldots, \rho_n^{(m-1)}\}$, associated with each cluster in $\lambda^{(m-1)}$.

We initialize the proposed values $\lambda^{(N)}$ and $\mathcal{R}^{(N)}$ by setting the latent partition and the unique values of the random orders equal to the previous state values, i.e. we set $\lambda^{(N)} = \lambda^{(m-1)}$ and $\mathcal{R}^{*(N)} = \mathcal{R}^{*(m-1)}$. The split and merge step starts by sampling two indices $i \neq \ell$ from $\{1, \ldots, n\}$. Then, one of the following cases can occur.

(a) If the observations i and ℓ belong to the same cluster, e.g. to the generic s block $B_s^{(m-1)}$ of $\lambda^{(m-1)}$, then we perform a split step.

- We update the proposed partition $\lambda^{(N)}$ according to the split step: we assign i to $B_s^{(N)}$ and ℓ to a new block $B_{k+1}^{(N)}$. We then randomly assign all the remaining observations in $B_s^{(m-1)} \setminus \{i, \ell\}$ to $B_S^{(N)}$ or $B_{k+1}^{(N)}$.
- We update the proposed unique values of the random orders $\mathcal{R}^{*(N)}$, whereas we first increase the dimension of $\mathcal{R}^{*(N)}$ to $k+1$ and then the elements s and $k+1$ are sampled from the mixture of posterior distributions in (B.4).
- Compute the acceptance rate

$$\alpha(\lambda^{(N)}, \mathcal{R}^{*(N)} \mid \lambda^{(m-1)}, \mathcal{R}^{*(m-1)}) = \min\left\{1, \frac{q(\lambda^{(m-1)}, \mathcal{R}^{*(m-1)} \mid \lambda^{(N)}, \mathcal{R}^{*(N)})}{q(\lambda^{(N)}, \mathcal{R}^{*(N)} \mid \lambda^{(m-1)}, \mathcal{R}^{*(m-1)})} \times \frac{\mathcal{L}(\lambda^{(N)})\mathcal{L}(\mathcal{R}^{*(N)})}{\mathcal{L}(\lambda^{(m-1)})\mathcal{L}(\mathcal{R}^{*(m-1)})} \frac{\mathcal{L}(\mathcal{Y} \mid \lambda^{(N)}, \mathcal{R}^{*(N)})}{L(\mathcal{Y} \mid \lambda^{(m-1)}, \mathcal{R}^{*(m-1)})}\right\}.$$

Thanks to the factorization in (B.3), the first ratio in the acceptance rate becomes

$$\frac{q(\lambda^{(m-1)} \mid \lambda^{(N)})}{q(\lambda^{(N)} \mid \lambda^{(m-1)})} \frac{q(\mathcal{R}^{(m-1)} \mid \mathcal{Y})}{q(\mathcal{R}^{(N)} \mid \mathcal{Y})} = 2^{|B_s^{(m-1)}|-2} \frac{\psi(\rho_s^{*(m-1)} \mid \mathcal{Y})}{\psi(\rho_s^{*(N)} \mid \mathcal{Y})\psi(\rho_{k+1}^{*(N)} \mid \mathcal{Y})}$$

where the first term follows the derivation in Jain and Neal (2004) while the second is given by the mixture of posteriors proposal distributions. For the remaining terms, the followings hold

$$\frac{\mathcal{L}(\lambda^{(N)})}{\mathcal{L}(\lambda^{(m-1)})} = \frac{\Gamma\big(\alpha + n_s^{(N)}\big)\Gamma\big(\alpha + n_{k+1}^{(N)}\big)}{\Gamma(\alpha)\Gamma\big(\alpha + n_s^{(m-1)}\big)}$$

$$\frac{\mathcal{L}(\mathcal{R}^{*(N)})}{\mathcal{L}(\mathcal{R}^{*(m-1)})} = \frac{1 - \prod_{j=0}^{k}\left(1 - \frac{1}{2^{T-1}-j}\right)}{1 - \prod_{j=0}^{k-1}\left(1 - \frac{1}{2^{T-1}-j}\right)}$$

$$\frac{\mathcal{L}(\mathcal{Y} \mid \lambda^{(N)}, \mathcal{R}^{*(N)})}{\mathcal{L}(\mathcal{Y} \mid \lambda^{(m-1)}, \mathcal{R}^{*(m-1)})} = \frac{\prod_{j \in B_s^{(N)}} \mathcal{L}\big(\boldsymbol{y}_j \mid \rho_s^{*(N)}\big) \prod_{j \in B_{k+1}^{(N)}} \mathcal{L}\big(\boldsymbol{y}_j \mid \rho_{k+1}^{*(N)}\big)}{\prod_{j \in B_s^{(m-1)}} \mathcal{L}\big(\boldsymbol{y}_j \mid \rho_s^{*(m-1)}\big)}$$

where $n_s^{(N)}$ and $n_s^{(m-1)}$ denote the size of the sth block in the proposed partition and the previous state, respectively, and the latter equation follows the restricted Gibbs sampling strategy discussed in Jain and Neal (2004).

(b) If observations i and ℓ belong to different clusters, e.g. $i \in B_s^{(m-1)}$ and $\ell \in B_r^{(m-1)}$ with $s < r$, then we perform a merge step.

- We update the proposed partition $\lambda^{(N)}$ according to the merge step: we assign all the observations in $B_r^{(m-1)}$ to $B_s^{(m-1)}$ and we destroy the block $B_r^{(m-1)}$.

- We update the proposed state for the unique values of the random orders $\mathcal{R}^{*(N)}$ by destroying the rth element and by sampling the sth element from the mixture of posterior distributions in (B.4).

- Compute the acceptance rate

$$\alpha(\lambda^{(N)}, \mathcal{R}^{*(N)} \mid \lambda^{(m-1)}, \mathcal{R}^{*(m-1)}) = \min\left\{1, \frac{q(\lambda^{(m-1)}, \mathcal{R}^{*(m-1)} \mid \lambda^{(N)}, \mathcal{R}^{*(N)})}{q(\lambda^{(N)}, \mathcal{R}^{*(N)} \mid \lambda^{(m-1)}, \mathcal{R}^{*(m-1)})} \times \frac{\mathcal{L}(\lambda^{(N)})\mathcal{L}(\mathcal{R}^{*(N)})}{\mathcal{L}(\lambda^{(m-1)})\mathcal{L}(\mathcal{R}^{*(m-1)})} \frac{\mathcal{L}(\mathcal{Y} \mid \lambda^{(N)}, \mathcal{R}^{*(N)})}{L(\mathcal{Y} \mid \lambda^{(m-1)}, \mathcal{R}^{*(m-1)})}\right\}.$$

Similarly to point (a), we have

$$\frac{q(\lambda^{(m-1)} \mid \lambda^{(N)})}{q(\lambda^{(N)} \mid \lambda^{(m-1)})} \frac{q(\mathcal{R}^{(m-1)} \mid \mathcal{Y})}{q(\mathcal{R}^{(N)} \mid \mathcal{Y})} = \left(\frac{1}{2}\right)^{|B_s^{(m-1)}|+|B_r^{(m-1)}|-2} \frac{\psi(\rho_s^{*(m-1)} \mid \mathcal{Y})\psi(\rho_r^{*(m-1)} \mid \mathcal{Y})}{\psi(\rho_s^{*(N)} \mid \mathcal{Y})}.$$

Further, for the remaining terms the followings hold

$$\begin{aligned}
\frac{\mathcal{L}(\lambda^{(N)})}{\mathcal{L}(\lambda^{(m-1)})} &= \frac{\Gamma(\alpha)\Gamma\big(\alpha + n_s^{(N)}\big)}{\Gamma\big(\alpha + n_s^{(m-1)}\big)\Gamma\big(\alpha + n_r^{()m-1}\big)}, \\
\frac{\mathcal{L}(\mathcal{R}^{*(N)})}{\mathcal{L}(\mathcal{R}^{*(m-1)})} &= \frac{1 - \prod_{j=0}^{k-1}\left(1 - \frac{1}{2^{T-1}-j}\right)}{1 - \prod_{j=0}^{k}\left(1 - \frac{1}{2^{T-1}-j}\right)}, \\
\frac{\mathcal{L}(\mathcal{Y} \mid \lambda^{(N)}, \mathcal{R}^{*(N)})}{\mathcal{L}(\mathcal{Y} \mid \lambda^{(m-1)}, \mathcal{R}^{*(m-1)})} &= \frac{\prod_{j \in B_s^{(N)}} \mathcal{L}\big(\boldsymbol{y}_j \mid \rho_s^{*(N)}\big)}{\prod_{j \in B_s^{(m-1)}} \mathcal{L}\big(\boldsymbol{y}_j \mid \rho_s^{*(m-1)}\big) \prod_{j \in B_r^{(m-1)}} \mathcal{L}\big(\boldsymbol{y}_j \mid \rho_{k+1}^{*(m-1)}\big)}.
\end{aligned}$$

Finally, we perform a Metropolis–Hasting step by sampling a uniform random variable $U \sim Unif(0,1)$. If $U < \alpha(\lambda^{(N)}, \mathcal{R}^{*(N)} \mid \lambda^{(m-1)}, \mathcal{R}^{*(m-1)})$ we accept the proposed values $\lambda^{(N)}$ and $\mathcal{R}^{*(N)}$ as current state of the chain, setting $(\lambda^{(m)}, \mathcal{R}^{*(m)})$ equal to $(\lambda^{(N)}, \mathcal{R}^{*(N)})$, otherwise if $U > \alpha(\lambda^{(N)}, \mathcal{R}^{*(N)} \mid \lambda^{(m-1)}, \mathcal{R}^{*(m-1)})$ we stay on the previous state of the chain and we set $(\lambda^{(m)}, \mathcal{R}^{*(m)})$ equal to $(\lambda^{(m-1)}, \mathcal{R}^{*(m-1)})$.

B.2 Additional details on the SIR model

The scaled stochastic process $\boldsymbol{X}^{(\varepsilon)}$ is a time-inhomogeneous continuous-time Markov process generated by (Gikhman and Skorokhod, 2004, Chapter 1) (see also Applebaum (2019); Engel and Nagel (2000))

$$\begin{aligned}
A_t^{(\varepsilon)} g(x) = {} & \varepsilon^{-1}\beta(t)x_1x_2(g(x_1 - \varepsilon, x_2 + \varepsilon, x_3) - g(x)) \\
& + \varepsilon^{-1}\xi(t)x_2(g(x_1, x_2 - \varepsilon, x_3 + \varepsilon) - g(x)),
\end{aligned} \tag{B.5}$$

for bounded, continuous functions $g : \mathbb{R}_+^3 \mapsto \mathbb{R}$. The trajectories of the scaled process $X^{(\varepsilon)}$ can be described as solutions to the following stochastic differential equations (written in the integral form)

$$\begin{aligned}
X_S^{(\varepsilon)}(t) &= X_S^{(\varepsilon)}(0) - \varepsilon \int_0^t \int_0^\infty \mathbb{1}_{[0,\varepsilon^{-1}\beta(u)X_S^{(\varepsilon)}(u_-)X_I^{(\varepsilon)}(u_-)]}(v)\, Q_1(\mathrm{d}u, \mathrm{d}v), \\
X_I^{(\varepsilon)}(t) &= X_I^{(\varepsilon)}(0) + \varepsilon \int_0^t \int_0^\infty \mathbb{1}_{[0,\varepsilon^{-1}\beta(u)X_S^{(\varepsilon)}(u_-)X_I^{(\varepsilon)}(u_-)]}(v)\, Q_1(\mathrm{d}u, \mathrm{d}v) \\
&\quad - \varepsilon \int_0^t \int_0^\infty \mathbb{1}_{[0,\varepsilon^{-1}\xi(u)X_I^{(\varepsilon)}(u_-)]}(v)\, Q_2(\mathrm{d}u, \mathrm{d}v), \\
X_R^{(\varepsilon)}(t) &= X_R^{(\varepsilon)}(0) + \varepsilon \int_0^t \int_0^\infty \mathbb{1}_{[0,\varepsilon^{-1}\xi(u)X_I^{(\varepsilon)}(u_-)]}(v)\, Q_2(\mathrm{d}u, \mathrm{d}v),
\end{aligned} \tag{B.6}$$

where Q_1, Q_2 are independent Poisson random measures on $\mathbb{R}^2$ with intensity measure $\mathrm{d}u \times \mathrm{d}v$ where $\mathrm{d}u$, and $\mathrm{d}v$ are Lebesgue measures on $\mathbb{R}$. We have used the notation $\mathbb{1}_A(x)$ to denote the indicator function of a set A, which takes the value one when $x \in A$, and zero otherwise. The process X (or the scaled process $X^{(\varepsilon)}$) can simulated as a pure jump process using standard techniques. An alternative individual-based approach is provided by what is known as the *Sellke construction* Andersson and Britton (2000). Let $H_\varepsilon(t)$ denote the cumulative infection pressure upto time t, where

$$H_\varepsilon(t) = \int_0^t \varepsilon\beta(u)X_I(u)\mathrm{d}u = \int_0^t \beta(u)X_I^{(\varepsilon)}(u)\mathrm{d}u. \tag{B.7}$$

According to the Sellke construction, each susceptible individual is assigned an exponentially distributed (with mean one) threshold E, and gets infected at time t as soon as $E \leq H_\varepsilon(t)$. If such a time never arrives, *i.e.*, if $E > \lim_{t\to\infty} H_\varepsilon(t)$ (the limit is finite almost surely), then the individual escapes infection. That is, conditional on the history of the process up to time t, the probability that a randomly chosen susceptible individual is still susceptible at time t, *i.e.*, has not been infected by time t, is $\exp(-H_\varepsilon(t))$. This individual-based perspective is at the heart of the Dynamical Survival Analysis (DSA) KhudaBukhsh et al. (2020); Di Lauro et al. (2022) approach to parameter inference for infectious disease epidemiology based on a random sample of infection times (and recovery times, if available) as opposed to standard methods that require (or impute when full data are not available) population-level trajectories.

To study the large population limit of the scaled stochastic process $X^{(\varepsilon)}$, let us rewrite the trajectory in (B.6) as

$$\begin{aligned}
X_S^{(\varepsilon)}(t) &= X_S^{(\varepsilon)}(0) - M_S^{(\varepsilon)}(t) - \int_0^t \beta(u)X_S^{(\varepsilon)}(u_-)X_I^{(\varepsilon)}(u_-)\mathrm{d}u,\\
X_I^{(\varepsilon)}(t) &= X_I^{(\varepsilon)}(0) + M_I^{(\varepsilon)}(t) + \int_0^t \beta(u)X_S^{(\varepsilon)}(u_-)X_I^{(\varepsilon)}(u_-)\mathrm{d}u - \int_0^t \xi(u)X_I^{(\varepsilon)}(u_-)\mathrm{d}u,\\
X_R^{(\varepsilon)}(t) &= X_R^{(\varepsilon)}(0) + M_R^{(\varepsilon)}(t) + \int_0^t \xi(u)X_I^{(\varepsilon)}(u_-)\mathrm{d}u,
\end{aligned}$$

where the stochastic processes $M_S^{(\varepsilon)}$, $M_I^{(\varepsilon)}$, and $M_R^{(\varepsilon)}$ are square-integrable zero-mean martingales defined as

$$\begin{aligned}
M_S^{(\varepsilon)}(t) &= \varepsilon \int_0^t \int_0^\infty \mathbb{1}_{[0,\varepsilon^{-1}\beta(u)X_S^{(\varepsilon)}(u-)X_I^{(\varepsilon)}(u-)]}(v)\,\bar{Q}_1(\mathrm{d}u,\mathrm{d}v),\\
M_I^{(\varepsilon)}(t) &= \varepsilon \int_0^t \int_0^\infty \mathbb{1}_{[0,\varepsilon^{-1}\beta(u)X_S^{(\varepsilon)}(u-)X_I^{(\varepsilon)}(u-)]}(v)\,\bar{Q}_1(\mathrm{d}u,\mathrm{d}v)\\
&\qquad - \varepsilon \int_0^t \int_0^\infty \mathbb{1}_{[0,\varepsilon^{-1}\xi(u)X_I^{(\varepsilon)}(u-)]}(v)\,\bar{Q}_2(\mathrm{d}u,\mathrm{d}v),\\
M_R^{(\varepsilon)}(t) &= \varepsilon \int_0^t \int_0^\infty \mathbb{1}_{[0,\varepsilon^{-1}\xi(u)X_I^{(\varepsilon)}(u-)]}(v)\,\bar{Q}_2(\mathrm{d}u,\mathrm{d}v),
\end{aligned}$$

where $\bar{Q}_1$ and $\bar{Q}_2$ are the compensated Poisson random measures (corresponding to Q_1 and Q_2 respectively). It is easy to verify that the quadratic variations of the above martingales all converge to zero in probability as $\varepsilon \to 0$, which in turn imply that the martingales themselves

converge to the deterministic function taking the value zero at all times. Consider the system of Ordinary Differential Equations (ODEs) in (3.14) with the initial condition $S(0) = 1, I(0) = I_0$, and $R(0) = 0$. Let $\bar{X} = (S, I, R)$. Then, note that $X^{(\varepsilon)}(0) \to (1, I_0, 0)$ as $\varepsilon \to 0$ by design, and as a consequence of the Functional Law of Large Numbers (FLLN), we can prove that

$$\lim_{\varepsilon \to 0} \sup_{t \leq T} ||X^{(\varepsilon)}(t) - \bar{X}(t)||_\infty \to 0 \tag{B.8}$$

in probability where $||(x_1, x_2, x_3)||_\infty = \max\{|x_1|, |x_2|, |x_3|\}$. In fact, the above convergence holds almost surely since we have assumed the functions $\beta(t)$ and $\xi(t)$ are bounded. The proof follows by standard arguments using Grönwall's inequality and some maximal estimates. See, for example, (KhudaBukhsh and Rempała, 2024, Theorem 3.1), where it is also discussed how an SIR model with time-varying infection and recovery rates such as the model we use in this paper can be used to approximate a compartmental Susceptible-Exposed-Infected-Recovered (SEIR) model.

B.3 Additional figures, tables and algorithms

In this section we first show in Figure B.1 an example of simulated data for the time series synthetic study. The data generating process and observation-specific parameters are described in Section 3.5. The figure shows in the left panel different time series, whereas each series is an observation characterized by its local parameters, but some of the series share the same structural change times. The right part of the figure shows two posterior similarity matrices, associated with two distinct estimates of the model, with the same accuracy of the normalisation constant B = 10 000, but varying the depth of the proposal, top with L = 1 and bottom with L = 100.

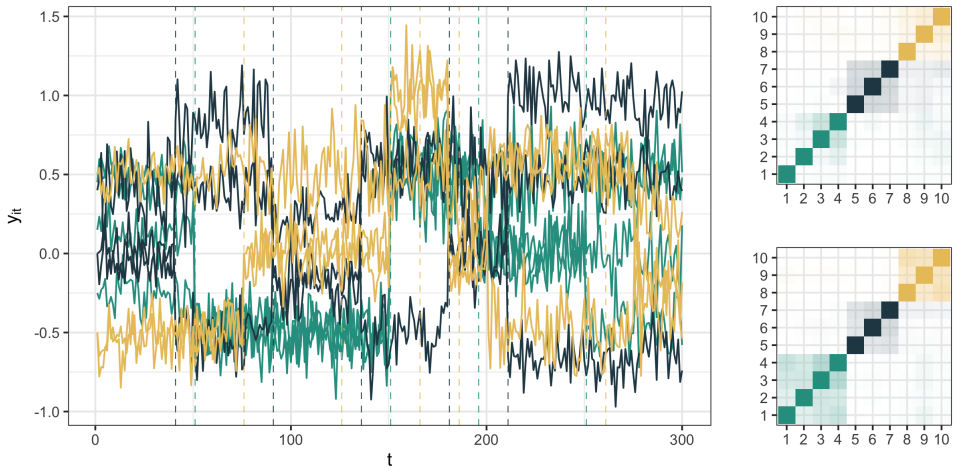


Figure B.1: Left panel: one realisation of simulated data according to the model in (3.9) and parameters in Table 3.1. Right panels: posterior similarity matrix of one replication with B = 10 000, top with L = 1 and bottom with L = 100 respectively.

Regarding the application to exchange rate data in Section 3.5.1, Figure B.2 reports the marginal change points of the five smallest estimated clusters.

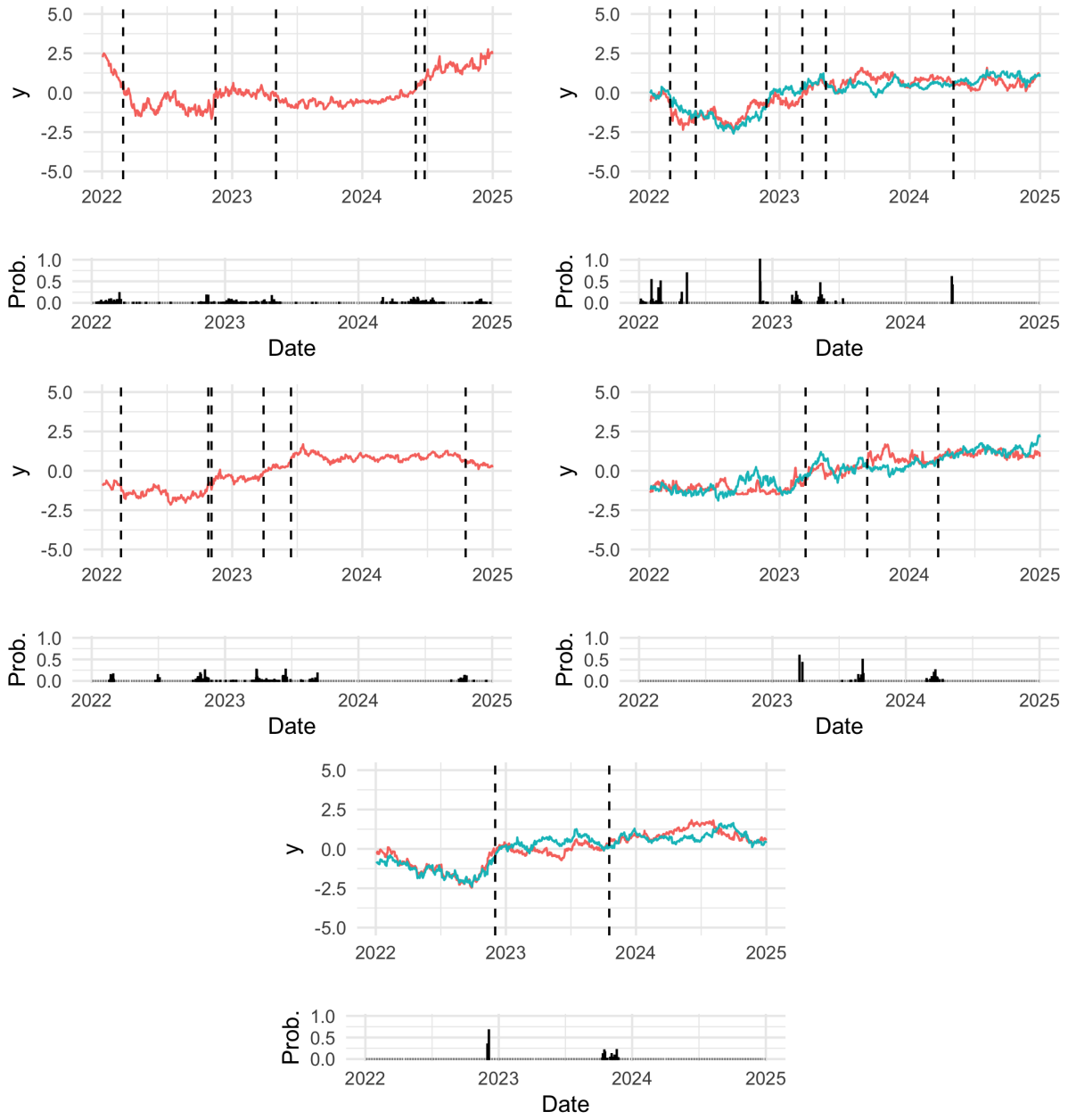


Figure B.2: Marginal change points for the five smallest clusters of the application on Euro exchange rates. In each block the histogram in the lower part represents the frequency of times that each time point has been detected as a change point in the MCMC chain, in the upper block are represented the exchange rates of each group with the marginal change points of the cluster denoted by the dashed lines.

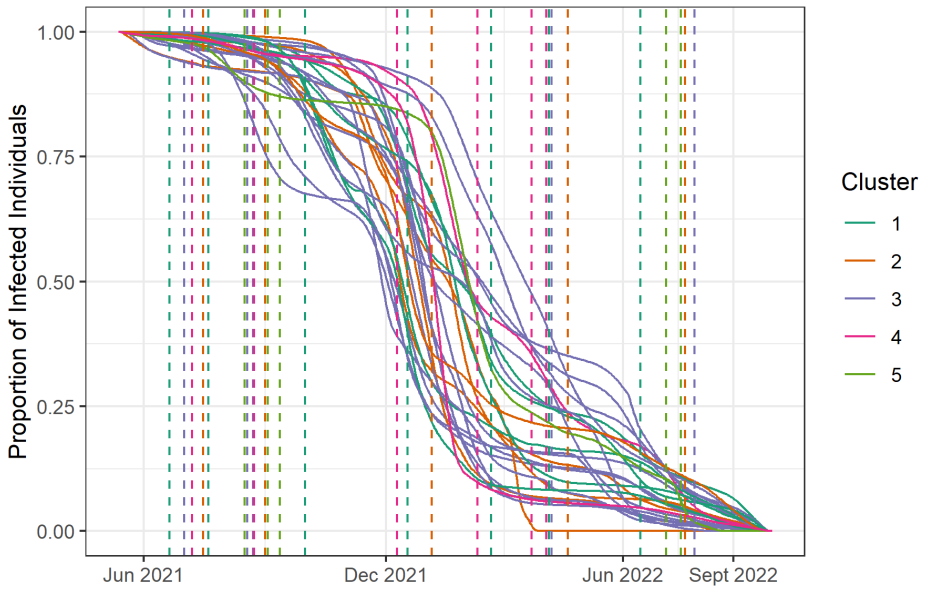


Figure B.3: COVID-19 empirical survival functions of the EU countries. Colors represent the group in the final clustering and the dashed lines the marginal change points of each cluster.

For the application to COVID-19 real data in Section 3.6.2, Figure B.3 shows the empirical survival function of each country colored according to the final clustering and the marginal change points that characterizes each cluster.

Sampling from (3.8) is unfeasible due to the intractability of the normalisation constant in each component of the mixture $\mathcal{L}(\rho \mid \boldsymbol{y}_i)$, $i = 1, \dots, n$. Normalising each component requires the computation of $\sum_{r=1}^{2^{T-1}} \mathcal{L}(\widetilde{\rho}_r \mid \boldsymbol{y}_i)$ for each observation $\boldsymbol{y}_i$, where $\{\widetilde{\rho}_1, \dots \widetilde{\rho}_{2^{T-1}}\}$ are possible orders of T elements. Since the number of elements explodes as far as T increases, evaluating the normalisation constants becomes soon unfeasible. We then proceed by pre-computing these constants using an importance sampling strategy: the importance distribution follows a Binomial-Multinomial law, for which we first generate the number of blocks k from a Binomial distribution and then the frequencies from a Multinomial distribution of dimension k. Then, we proceed with a standard importance reweighting approach. Algorithm 1 shows an implementation of such strategy for a single normalisation constant.

Algorithm 1: Computation of the ith normalisation constant of (3.8)

input number of sampled orders B for the importance sampling (accuracy) and success probability $p \in (0,1)$ of the binomial distribution.

for $r \in 1, \ldots, \mathrm{B}$ **do**

a) **sample** $k \sim \text{Binomial}(T-1,\, p)$.

b) **sample** $\left(|B_1^{(r)}|, \ldots, |B_k^{(r)}|\right)^\top \sim Multinom(T,\, (1/k, \ldots, 1/k)^\top)$ and set $\rho_r^\dagger = \{B_1^{(r)}, \ldots, B_k^{(r)}\}$.

d) **compute** $\phi(\rho_r^\dagger) = \binom{T}{k} p^k (1-p)^{T-k}\, T! \prod_{i=1}^{k} \frac{1}{\left(k^{|B_i^{(r)}|}\right)\left(|B_i^{(r)}|!\right)}$.

compute the approximation of $\sum_{r=1}^{2^{T-1}} \mathcal{L}(\widetilde{\rho}_r \mid \boldsymbol{y}_i) \simeq \frac{2^{T-1}}{\mathrm{B}} \sum_{r=1}^{\mathrm{B}} \frac{1}{\phi(\rho_r^\dagger)} \mathcal{L}(\rho_r^\dagger \mid \boldsymbol{y}_i)$.

end

Finally, Algorithm 2 shows the pseudo-code for producing a sample of infection times in a fixed time window, bounded by the last observational time T. Here we consider an algorithm that works with a functional infection rate $\beta(u)$, which in our case is obtained as step function starting from a vector of local infection rates $\boldsymbol{\beta}$.

Algorithm 2: Doob-Gillespie algorithm to sample infection times

input a functional $\beta(u)$ of time-varying infection rates and a value ξ for the recovery rate, starting values for S_0, I_0 and R_0.

denote with (S_t, I_t, R_t) the vectors of susceptible, infected and recovered individuals at time t.

set $t = 0$ and $S_t = S_0$, $I_t = I_0$, $R_t = R_0$.

while $t < T$, $I_t > 0$ *and* $S_t > 0$ **do**

a) **sample** E_1 from $P(E_1 > e_1) = \exp\left(-\frac{S_t I_t}{S_0} \int_t^{t+e_1} \beta(u)du\right)$ and $E_2 \sim \text{Exp}\,(\xi I_t)$.

b) **set** $t^* = \min(E_1, E_2)$.

- **if** $t^* = E_1$, then set $S_t = S_t - 1$, $I_t = I_t + 1$, $R_t = R_t$, and $\delta = 1$.
- **else if** $t^* \neq E_1$, then set $S_t = S_t$, $I_t = I_t - 1$, $R_t = R_t + 1$, and $\delta = 0$.

c) Set $t = t + t^*$. If $\delta = 1$ **then** save t as new infection time.

end

i	$\{\beta_1^*, \beta_2^*\}$	I_0	$\{m_{i,1}, m_{i,2}\}$
1	{0.211, 0.55}	23/S0	{120, 80}
2	{0.221, 0.50}	23/S0	
3	{0.218, 0.54}	21/S0	
4	{0.225, 0.51}	20/S0	
5	{0.213, 0.52}	24/S0	{70, 130}
6	{0.213, 0.51}	23/S0	
7	{0.193, 0.57}	22/S0	
8	{0.195, 0.54}	21/S0	{30, 170}
9	{0.191, 0.53}	20/S0	
10	{0.189, 0.51}	24/S0	

Table B.1: Parameters of the data generating processes for the epidemiological synthetic study. Left to right: observation index, different local infection rates for each series, starting infection rate and true latent order shared among observations in the same cluster.

Appendix C

Chapter 4 supplementary materials

C.1 Algorithms

Here we show the scheme of the algorithms implemented in **BayesChange** for change points detection and clustering. Algorithm 3 shows the procedure for detecting change points as presented in Section 4.2.1, this is included in C++ functions `detect_cp_uni`, `detect_cp_multi` and `detect_cp_epi`. Algorithm 4 is the procedure for clustering time series and epidemic diffusions with common change points as detailed in Section 4.2.2, and implemented in C++ functions `clust_cp_uni`, `clust_cp_multi` and `clust_cp_epi`.

Algorithm 3: Split and merge algorithm in to detect change points in `detect_cp`.

input a starting order $\rho_i^{(0)} = (A_{i,1}^{(0)}, \ldots, A_{i,m_i}^{(0)})$ of $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iT})$, the number $M > 0$ of MCMC iterations and $q \in (0, 1)$.

for $m = 1, \ldots, M$ **do**

a) **set** $\rho_i^{(N)} = \rho_i^{(m-1)}$, with $(A_{i,1}^{(N)}, \ldots, A_{i,m_i}^{(N)})$ denoting the blocks of $\rho_i^{(N)}$.

b) **with probability** q perform a **split**:

- randomly choose one block in $(A_{i,1}^{(N)}, \ldots, A_{i,m_i}^{(N)})$;
- sample two consecutive observations and split the chosen block in two new groups and define $\rho_i^{(N)} = (A_{i,1}^{(N)}, \ldots, A_{i,m_i+1}^{(N)})$;
- evaluate the proposal, if accepted $\rho_i^{(m)} = \rho_i^{(N)}$, otherwise $\rho_i^{(m)} = \rho_i^{(m-1)}$.

otherwise perform a **merge**:

- randomly choose one block in $(A_{i,1}^{(N)}, \ldots, A_{i,m_i}^{(N)})$;
- merge the block with the consecutive one and define $\rho_i^{(N)} = (A_{i,1}^{(N)}, \ldots, A_{i,m_i-1}^{(N)})$;
- evaluate the proposal, if accepted $\rho_i^{(m)} = \rho_i^{(N)}$, otherwise $\rho_i^{(m)} = \rho_i^{(m-1)}$.

c) **If** $m_i > 1$ perform a **shuffle**:

- set $\rho_i^{(N)} = \rho_i^{(m)}$
- randomly choose one block $A_{i,j}^{(N)}$ in $\rho_i^{(N)}$;
- rearrange randomly the observations of $A_{i,j}^{(N)}$ and $A_{ij+1}^{(N)}$;
- evaluate the proposal, if accepted $\rho_i^{(m)} = \rho_i^{(N)}$, otherwise $\rho_i^{(m)} = \rho_i^{(m)}$.

d) **update** parameters σ, δ and ϕ.

end

Algorithm 4: Split and merge algorithm that updates λ in `clust_cp`.

input a partition $\lambda^{(0)} = \{B_1^{(0)}, \ldots, B_k^{(0)}\}$ of $\mathcal{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$, initial values for the unique latent orders $\mathcal{R}^{*(0)}$ and the number $M > 0$ of MCMC iterations.

for $m = 1, \ldots, M$ **do**

a) **set** $\lambda^{(N)} = \lambda^{(m-1)}$, with $\{B_1^{(N)}, \ldots, B_k^{(N)}\}$ denoting the blocks of $\lambda^{(N)}$, and $\mathcal{R}^{*(N)} = \mathcal{R}^{*(0)}$.

b) **sample** $i, \ell \in \{1, \ldots, n\}$ such that $i \neq \ell$.

if both i and ℓ belong to the same block $B_s^{(m-1)}$, perform a split:

i) assign i to $B_s^{(N)}$ and ℓ to a new block $B_{k+1}^{(N)}$;

ii) assign randomly each values of $B_s^{(m-1)}$ to $B_s^{(N)}$ or $B_{k+1}^{(N)}$;

iii) sample the two distinct unique values of the latent orders in $\mathcal{R}^{*(N)}$ associated with the observations whose indices belong to $B_s^{(N)}$ or $B_{k+1}^{(N)}$ from (3.8).

else if i and ℓ belong to different blocks, with $i \in B_s^{(m-1)}$ and $\ell \in B_w^{(m-1)}$, perform a merge:

i) assign all the indices in $B_w^{(m-1)}$ to $B_s^{(m-1)}$ and destroy $B_w^{(m-1)}$;

ii) sample the unique value of the latent orders in $\mathcal{R}^{*(N)}$ associated with all the observations whose indices are in $B_s^{(m-1)}$ from (3.8).

c) **perform** a Metropolis–Hastings step to accept the proposed values:

$\rightarrow$ if accepted, set $(\lambda^{(m)}, \mathcal{R}^{*(m)}) = (\lambda^{(N)}, \mathcal{R}^{*(N)})$,

$\rightarrow$ otherwise, set $(\lambda^{(m)}, \mathcal{R}^{*(m)}) = (\lambda^{(m-1)}, \mathcal{R}^{*(m-1)})$.

d) **update** the unique values $\rho_1^{*(m)}, \ldots, \rho_k^{*(m)}$ in $\mathcal{R}^{*(m)}$.

end

C.2 Code

The following is the code for generating synthetic univariate time series for the clustering application in Section 4.4.2.

```
R> data <- matrix(NA, nrow = 5, ncol = 200)
R> data[1, 1] <- rnorm(n = 1, mean = 0, sd = 0.100)
R> data[2, 1] <- rnorm(n = 1, mean = 0, sd = 0.125)
R> data[3, 1] <- rnorm(n = 1, mean = 0, sd = 0.175)
R> for(i in 2:50){
R>  data[1, i] <- 0.1 * data[1, i-1] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.100)
R>  data[2, i] <- 0.1 * data[2, i-1] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.125)
R>  data[3, i] <- 0.1 * data[3, i-1] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.110)
R> }
```

```
R> data[1, 51] <- rnorm(1, mean = 1, sd = 0.230)
R> data[2, 51] <- rnorm(1, mean = 1, sd = 0.225)
R> data[3, 51] <- rnorm(1, mean = 1, sd = 0.240)
R> for(i in 52:150){
R>  data[1, i] <- 0.1 * data[1, i-1] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.230)
R>  data[2, i] <- 0.1 * data[2, i-1] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.225)
R>  data[3, i] <- 0.1 * data[3, i-1] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.240)
R>}
R> data[1, 151] <- rnorm(1, mean = 0.5, sd = 0.225)
R> data[2, 151] <- rnorm(1, mean = 0.5, sd = 0.235)
R> data[3, 151] <- rnorm(1, mean = 0.5, sd = 0.100)
R> for(i in 152:200){
R>  data[1, i] <- 0.1 * data[1, i-1] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.225)
R>  data[2, i] <- 0.1 * data[2, i-1] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.235)
R>  data[3, i] <- 0.1 * data[3, i-1] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.100)
R> }
R> data[4, 1] <- rnorm(1, mean = 0, sd = 0.135)
R> data[5, 1] <- rnorm(1, mean = 0, sd = 0.155)
R> for(i in 2:25){
R>  data[4, i] <- 0.1 * data[4,i-1] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.135)
R>  data[5, i] <- 0.1 * data[5,i-1] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.155)
R> }
R> data[4, 26] <- rnorm(1, mean = 1, sd = 0.165)
R> data[5, 26] <- rnorm(1, mean = 1, sd = 0.185)
R> for(i in 27:200){
R>  data[4, i] <- 0.1 * data[4, i-1] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.165)
R>  data[5, i] <- 0.1 * data[5, i-1] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.185)
R> }
```

The following is the code for generating synthetic multivariate time series for the clustering application in Section 4.4.2.

```
R> data <-  array(data = NA, dim = c(2, 200, 5))
R> data[1:2,1,1] <- rnorm(1 ,mean = 0, sd = 0.100)
```

```
R> data[1:2,1,2] <- rnorm(1, mean = 0, sd = 0.125)
R> data[1:2,1,3] <- rnorm(1, mean = 0, sd = 0.175)
R> for(i in 2:50){
R>  data[1, i, 1] <- 0.1 * data[1, i-1, 1] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.100)
R>  data[2, i, 1] <- 0.1 * data[2, i-1, 1] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.100)
R>  data[1, i, 2] <- 0.1 * data[1, i-1, 2] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.125)
R>  data[2, i, 2] <- 0.1 * data[2, i-1, 2] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.125)
R>  data[1, i, 3] <- 0.1 * data[1, i-1, 3] + (1 - 0.1) * 0 +
R>      rnorm(1 ,mean = 0, sd = (1 - 0.1^2) * 0.110)
R>  data[2, i, 3] <- 0.1 * data[2, i-1, 3] + (1 - 0.1) * 0 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.110)
R> }
R> data[1, 51, 1] <- data[2, 51, 1] <- rnorm(1, mean = 1, sd = 0.230)
R> data[1, 51, 2] <- data[2, 51, 2] <- rnorm(1, mean = 1, sd = 0.225)
R> data[1, 51, 3] <- data[2, 51, 3] <- rnorm(1, mean = 1, sd = 0.240)
R> for(i in 52:150){
R>  data[1, i, 1] <- 0.1 * data[1, i-1, 1] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.230)
R>  data[2, i, 1] <- 0.1 * data[2, i-1, 1] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.230)
R>  data[1, i, 2] <- 0.1 * data[1, i-1, 2] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.225)
R>  data[2, i, 2] <- 0.1 * data[2, i-1, 2] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.225)
R>  data[1, i, 3] <- 0.1 * data[1, i-1, 3] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.240)
R>  data[2, i, 3] <- 0.1 * data[2, i-1, 3] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.240)
R> }
R> data[1:2, 151, 1] <- rnorm(2, mean = 0.5, sd = 0.225)
R> data[1:2, 151, 2] <- rnorm(2, mean = 0.5, sd = 0.235)
R> data[1:2, 151, 3] <- rnorm(2, mean = 0.5, sd = 0.100)
R> for(i in 152:200){
R>  data[1, i, 1] <- 0.1 * data[1, i-1, 1] + (1 - 0.1) * 0.5 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.225)
R>  data[2, i, 1] <- 0.1 * data[2, i-1, 1] + (1 - 0.1) * 0.5 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.225)
R>  data[1, i, 2] <- 0.1 * data[1, i-1, 2] + (1 - 0.1) * 0.5 +
```

```
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.235)
R>  data[2, i, 2] <- 0.1 * data[2, i-1, 2] + (1 - 0.1) * 0.5 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.235)
R>  data[1, i, 3] <- 0.1 * data[1, i-1, 3] + (1 - 0.1) * 0.5 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.100)
R>  data[2, i, 3] <- 0.1 * data[2, i-1, 3] + (1 - 0.1) * 0.5 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.100)
R> }
R> data[1:2,1,4] <- rnorm(1, mean = 1, sd = 0.135)
R> data[1:2,1,5] <- rnorm(1, mean = 1, sd = 0.155)
R> for(i in 2:25){
R>  data[1, i, 4] <- 0.1 * data[1, i-1, 4] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.135)
R>  data[2, i, 4] <- 0.1 * data[2, i-1, 4] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.135)
R>  data[1, i, 5] <- 0.1 * data[1, i-1, 5] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.155)
R>  data[2, i, 5] <- 0.1 * data[2, i-1, 5] + (1 - 0.1) * 1 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.155)
R> }
R> data[1:2, 26, 4] <- rnorm(n = 1, mean = 0.5, sd = 0.165)
R> data[1:2, 26, 5] <- rnorm(n = 1, mean = 0.5, sd = 0.185)
R> for(i in 27:200){
R>  data[1, i, 4] <- 0.1 * data[1, i-1, 4] + (1 - 0.1) * 0.5 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.165)
R>  data[2, i, 4] <- 0.1 * data[2, i-1, 4] + (1 - 0.1) * 0.5 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.165)
R>  data[1, i, 5] <- 0.1 * data[1, i-1, 5] + (1 - 0.1) * 0.5 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.185)
R>  data[2, i, 5] <- 0.1 * data[2, i-1, 5] + (1 - 0.1) * 0.5 +
R>      rnorm(1, mean = 0, sd = (1 - 0.1^2) * 0.185)
R> }
```

Appendix D

Chapter 5 supplementary materials

D.1 Prior sampling

The following algorithm shows the procedure for sampling from the prior distribution of 5.3.

Algorithm 5: Metropolis–within–Gibbs update for the set of orders $\lambda = \{\rho_1, \ldots, \rho_k\}$.

input number of orders k, number of time points T, and probability $p \in (0, 1)$.

initialize a set of k distinct random orders $\lambda^{(0)} = \{\rho_1^{(0)}, \ldots, \rho_k^{(0)}\}$.

for $j \in 1, \ldots, k$ **do**

a) **generate** a candidate $\rho_j^{(1)}$ from a Binomial–Multinomial scheme:

i) **sample** $m_j^{(1)} \sim \text{Binomial}(T-1, p)$.

ii) **sample** $\rho_j^{(1)} \sim \text{Multinomial}\left(T, \left(\frac{1}{m_j^{(1)}}, \ldots, \frac{1}{m_j^{(1)}}\right)\right)$.

b) **set** $\rho_j^{(0)} = \rho_j^{(1)}$ with probability $\min(1, \beta_j)$, where

$$\beta_j = \frac{\binom{T}{m_j^{(1)}} p^{m_j^{(1)}} (1-p)^{T-m_j^{(1)}} T! \prod_{i=1}^{m_j^{(1)}} \left((m_j^{(1)})^{n_j^{(1)}} n_j^{(1)}!\right)^{-1}}{\binom{T}{m_j^{(0)}} p^{m_j^{(0)}} (1-p)^{T-m_j^{(0)}} T! \prod_{i=1}^{m_j^{(0)}} \left((m_j^{(0)})^{n_j^{(0)}} n_j^{(0)}!\right)^{-1}} \overline{\omega},$$

$$\overline{\omega} = \prod_{l \in [k] \setminus \{j\}} \frac{g(d(\rho_j^{(1)}, \rho_l^{(0)}))}{g(d(\rho_j^{(0)}, \rho_l^{(0)}))} = \prod_{l \in [k] \setminus \{j\}} \frac{\exp\left[-\tau\{d(\rho_j^{(1)}, \rho_l^{(0)})\}^{-\nu}\right]}{\exp\left[-\tau\{d(\rho_j^{(0)}, \rho_l^{(0)})\}^{-\nu}\right]}.$$

end