Bayesian Nonparametric Methods for Complex Datasets

Alex Ziyu Jiang

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:

Abel Rodriguez, Chair

Jon Wakefield, Chair

Adrian Dobra

Program Authorized to Offer Degree:
Statistics

University of Washington

Abstract

Bayesian Nonparametric Methods for Complex Datasets

Alex Ziyu Jiang

Co-Chairs of the Supervisory Committee:
Abel Rodriguez
Department of Statistics

Jon Wakefield
Department of Statistics and Biostatistics

Modern data tend to present complex structures that challenge classical modeling assumptions and frameworks, including heterogeneity and spatial and/or temporal dependency. Bayesian nonparametric (BNP) models are a powerful tools that can address these challenges. They enable flexible modeling structures that adapt to the data complexity and provides uncertainty estimation. This dissertation proposes several BNP methods that are applicable to a wide range of statistical learning problems in regression, clustering and density estimation, with applications in fields including global health and financial econometrics. In Chapter 2, we proposed a novel model that integrates the Bayesian additive regression tree prior (BART) into the Gaussian process spatial model, aimed at spatial prediction problems where the covariate effects may be nonlinear and flexible. In Chapter 3, we studied and compared the computational performance for multivariate Hawkes processes (MHP) models, a temporal processes commonly used to model mutually exciting behaviors in temporal event sequences. In Chapter 4, we apply the dependent Dirichlet process (DDP) to model the temporal dynamics in the MHP models. Our model allows for flexible and adaptive modeling for excitation functions while borrowing information across dimensions. Future research directions related to the topic of this dissertation is outlined in Chapter 5.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

Table Number Page

ACKNOWLEDGMENTS

I am beyond grateful and forever indebted to my PhD advisors, Abel Rodriguez and Jon Wakefield, for their incredible guidance, support, and constant trust in me throughout the progression of my PhD. I have learned so much from them as outstanding researchers, mentors, and people. I also want to extend my gratitude to Adrian Dobra for serving as my reading and dissertation committee member, and for giving me the valuable suggestions that contributed to this dissertation. I want to thank Wei Sun for funding my RA throughout a huge part of my PhD, and for being such a role model in shaping up my career as a budding researcher.

I would love to thank all the graduate students and postdocs in our department and the biostatistics department, especially Anupreet Porwal, Rayleigh Lei, Erin Lipman, Daniel Suen, Zhining Sui and Rui Wang, for their valuable discussions and helpful feedback on my research and also for providing emotional support that helps me stay sane throughout my PhD. I also want to thank the students in my cohort: James Buenfil, Apara Venkat, Shreya Prakash, Jessica Kunke, Ronak Mehta, Kenny Zhang, Trinity Fan and Jillian Fisher. We entered the program during the start of COVID-19 and weathered the uncertainties of endless zoom meetings - I would not be able to survive without the constant support from my cohort. I want to thank my college best friend Yuhan Xie for always being there for me, providing me with invaluable emotional support throughout the stressful days and sharing feedback on my practice talks and slide drafts.

I also want to thank the independent movie theaters and live music venues in Seattle for breathing the brief but much needed fresh air into my life between the long research hours. I want to thank Lorde, Mitski, SZA, Adrianne Lenker, Fiona Apple, Joanna Newsom, Erykah Badu and Lana Del Rey, whose albums I play all the time and accompanied me through countless nights spent doing research.

DEDICATION

This dissertation is dedicated to my parents, Wen Tong and Tiejun Jiang, who always believed in and supported me, through the thick and thin of our lives.

Chapter 1

INTRODUCTION

Modern data often demonstrate complex structures that challenge classical statistical modeling assumptions and frameworks. Examples include spatial data collected via stratified survey designs, temporal event sequence data that demonstrated mutually-exciting cross-dimensional dynamics, and data from heterogeneous populations with underlying structures. Traditional statistical methods tend to rely on stringent modeling assumptions that ignore or downplay many of those features. Thus, it is desirable to establish flexible modeling frameworks that allow for uncertainty quantification while taking into consideration the latent data dependencies and generic distributional structures, with adequate uncertainty estimates.

Bayesian nonparametric (BNP) models provide a powerful approach to these challenges. Contrary to parametric modeling where the model prior is placed on a finite-dimensional parameter space, BNP models are Bayesian models defined on an infinte-dimensional parameter space. For example, in a regression problem, we are interested in modeling the regression function of the model, and the parameter space can be the set of all regression functions. On the other hand, in a density estimation problem, the parameter space can correspond to the set of all density functions. As one of its special features, the number of parameter dimensions used by a BNP model is adaptive to the data, allowing the complexity of the model to grow as the size of the dataset increases.

The Gaussian processes prior [144] prior is widely used on regression functions. The properties of the regression function such as its smoothness and periodicity are controlled by its Gaussian kernel function. The Bayesian Additive Regression Trees (BART) [29] prior provides an alternative approach, under which the regression function is modeled as an additive sum of regression trees. Compared to the GP, BART controls the shape of the regression function by directly placing prior on the structure of the regression trees and

their node values.

For clustering problems, BNP models like Dirichlet process (DP) [85] prior place a prior distribution on the set of mixing measures in a mixture model, which controls how observations are partitioned into different clusters. Such nonparametric mixtures can also be applied to probability density estimation problems.

In the past, BNP models have been applied to various modeling problems with data that are complex. For example, Gaussian processes have been applied to study the modeling of the point-level spatial data [122, 34]. The Dirichlet process mixture has been combined with the Hawkes process [67] to study the clustering structure in temporal event sequences [44, 164]. While BNP models such as GP, BART and DP provide a framework for flexibly and coherently modeling the complex data dynamics including heterogeneity, clustering and nonlinearity, there are two major remaining challenges in the recent literature.

The first challenge is related to capturing the dependence in modeling. BNP models, while flexible by nature, can fail to appropriately capture the data dependence, without carefully consideration of the model. For example, while the BART method has been applied to spatial predictions [116, 89, 87], these models tend to assume variance-covariance structures that are very simple. Under such models, the spatial dependence in the data, which is usually modelling using Gaussian processes, may not be fully recovered. Another example is related to modeling the excitation kernels in multivariate Hawkes process models (MHP, [95]). Current methods tend to model these kernels independently, thereby disregarding the similarity shared across different dimensions, as pointed out by [140]. The second challenge is related to computation. BNP models tend to be computationally intensive [114], motivating the need to develop scalable computation tools that provide estimation, inference and prediction efficiently. This is especially important when these models are applied to real-world settings, where large-scaled datasets are collected under complicated designs.

This dissertation aims to address these challenges. Specifically, this dissertation develops several BNP models for datasets with complex structures and designs, including point and areal-level spatial data, temporal event sequences and data with hierarchical clustering structure. Under circumstances where BNP modeling is faced with computational challenges, the dissertation proposes several approximation methods that balance

model flexibility and scalability, including Laplacian [142] and variational approximation methods [13]. Combining methodological rigor with practical applicability, the dissertation applies the models to several real-world examples in fields including public health and finance, which are detailed in the following chapters:

- Chapter 2 proposes BARTSIMP, a novel model that integrates BART into Gaussian process spatial model. BARTSIMP is aimed at spatial prediction problems where the covariate effects tend to be nonlinear and flexible. To improve the computational scalability of the model, we incorporated the Integrated Nested Laplace Approximation (INLA) method into the MCMC update routine. The model is studied through simulation examples and applied to anthropometric survey data from Kenya, where the data was collected from a stratified two-stage sampling design.

- Chapter 3 addresses the computational challenges in conducting Bayesian inference for multivariate Hawkes processes (MHPs) [67, 95], a temporal point processes commonly used to capture mutually-exciting behaviors in temporal event sequences. Three stochastic gradient-based Bayesian inference algorithms are studied and compared, including the stochastic gradient expectation-maximization (SGEM), stochastic gradient variational inference (SGVI), and stochastic gradient Langevin dynamics (SGLD). A novel approximation to the likelihood function is also proposed that improves boundary behavior. The methods are compared via simulation studies and applied to theintraday Standard & Poor's 500 sector data to study the risk dynamics.

- As an extension to the previous chapter, Chapter 4 proposes MHP-DDP, a semiparametric model that models the excitation functions based on a dependent Dirichlet process. MHP-DDP allows for flexible and adaptive modeling for excitation functions while borrowing information across dimensinos. Two computation methods including MCMC and stochastic variational inference are developed. The modeling behavior of MHP-DDP is studied via simulation studies, where MHP-DDP outperform benchmarks in terms of estimation accuracy. MHP-DDP is applied to study the order flow

of high-frequency financial markets, and provided valuable insights to understand the excitation patterns in real order flow data.

- Chapter 5 describes future directions that are suggested by the contents of the dissertation.

Chapter 2

BARTSIMP: FLEXIBLE SPATIAL COVARIATE MODELING AND PREDICTION USING BAYESIAN ADDITIVE REGRESSION TREES

2.1 Introduction

There has been an increased interest in using covariates for spatial data modeling [156, 100], with previous studies showing that the inclusion of influential spatial covariates can lead to improved prediction accuracy [94, 165]. It is straightforward to include covariates in a linear model, within the linear predictor of spatial random effects models, but the extension to more flexible covariate models is more difficult. Flexible modeling is desirable to leverage nonlinearities and interactions which can lead to improved predictive performance.

Currently, Gaussian random field (GRF) models are commonly used as a modeling framework for capturing spatial correlations, with wide application, for example, in population health modeling [41]. For data that arise from complex survey designs, fully Bayesian modeling approaches have been explored, see for example [26, 9] and [50]. A common approach to computation in spatial modeling is Integrated nested Laplace approximation (INLA) [142], particularly in a low-and-middle-income countries (LMIC) context [155, 24]. INLA is a powerful tool for carrying out Bayesian inference, but requires the predictors to have a linear form, which cannot capture multivariate nonlinearities, including interactions, which may lead to reduced predictive performance. We do note that it is simple to include spline terms, since such models can be expressed in linear form, and also random walk models, both of which can simply capture univariate nonlinearities.

There are many machine learning methods that allow for flexible covariate modeling, many of which have been applied to describing variation in health and demographic indicators , see for example [19] and [23]. Particular methods that have been utilized include the convolutional neural network [77], classification trees [12], stacked generalization [35, 124] and random forests [132]. However, these approaches suffer from drawbacks. A number of

machine learning approaches in spatial modeling ignore the spatial dependence [56] while others (including the aforementioned references) either avoid giving uncertainty intervals or do not correctly propagate uncertainty, as is required for valid inference [36, 159].

Bayesian Additive Regression Trees (BART) [30] provide reliable Bayesian inference by specifying a prior distribution on the 'sum-of-trees' structure that flexibly models the covariates. Previous applications of BART on spatial data modeling problems include [116], who proposed a spatial BART model based on a conditional autoregressive (CAR) model, and [87], who suggested using a matrix exponential spatial specification [89] as an extension of CAR. [151] review extensions to the basic BART model, including a description of the approach due to [116]. However, compared to the GRF that we use, these models specify a simple variance-covariance structures and are designed for area-level data. The model considered by [146] assumes a more generic covariance structure under a non-spatial setting, but is limited to models with few random effects and does not scale well to large spatial datasets. As a result, we note that the large numbers of random effects in continuous spatial models, along with the strong dependence in the posterior, lead to computational difficulties and pose serious challenges for incorporating BART into spatial models.

In this chapter, we propose **BARTSIMP**, which is shorthand for '**BART** for **S**patial **I**NLA **M**odeling and **P**rediction', as a GRF spatial Bayesian modeling framework with a flexible covariate regression model. Our model leverages the flexibility in BART to capture the nonlinearity and interactions across covariates, while also recognizing the complex spatial correlation structure in the residuals which is modeled by the GRF. To ease computation, we use the INLA-within-MCMC method [61] to design a Metropolis-within-Gibbs sampler that integrates out the random effects using INLA and then performs MCMC updates on the remaining parameters. This model is the first to simultaneously model the covariate effects through BART and the spatial effects through a GRF, while producing Bayesian uncertainty estimates (credible intervals). Our spatial BART model is also the first to use INLA as an approximate approach to reduce the computational burden.

We organize Chapter 2 as follows. We will describe our motivating example, which is spatial prediction for child anthropometric data, in Section 2.2. The model formulation is in Section 2.3 and we describe the Metropolis-within-Gibbs sampler which we use for

model implementation in Section 2.4. We apply our model to simulated data experiments in Section 2.5 and return to the Kenya data in Section 2.7. Section 2.8 concludes Chapter 2 with a discussion.

2.2 Motivating Dataset

Our study is motivated by child undernutrition data collected in the 2014 Kenya Demographic and Health Survey (DHS). Specifically, we are interested in wasting for children under the age of five, which is measured using the weight-for-height Z-score (WHZ) metric. The Z-score can be interpreted as the number of units that the weight for a child is higher or lower than the median value, compared to all children of the same height in the population [106]. For example, wasting is defined as a WHZ score below -2 [81].

The data is collected via stratified two-stage cluster sampling. In the first stage, 1584 enumeration areas (EAs) were sampled across 92 strata. The 92 sampling strata are defined according to the 47 counties and the urban-rural status of the county (with Nairobi and Mombasa counties only having urban areas). At the second stage, 25 households were selected from each of the EAs, to give 40,300 households in total and yielding 20,977 child WHZ measurements.

We use geospatial covariates on the raster level, with a list of covariate descriptions in Table 2.1. In our example, for simplicity, we did not include the urban-rural status as a covariate in our model, which may lead to some bias due to oversampling of urban clusters but we believe that the inclusion of population density provides some protection.

The aim of the analysis is to provide predictions of WHZ at various administrative levels. To introduce some terminology, Admin 0 denotes the national level, Admin 1 one below that (for example, states in the United States) and Admin 2 one below that (for example, counties in the United States). In Kenya, the Admin 1 level contains 47 counties, while Admin 2 contains 290 constituencies. Hence, our objective is a problem in small area estimation (SAE). If there are sufficient data in each area, a weighted estimator can be used. Such estimates account for the design, but estimates in each area are based on data from that area only (and in the SAE literature are referred to as direct estimates) But often there are insufficient area-based data and models must be introduced. In an area-based

Table 2.1: DHS Geospatial covariates used in our model.

Variable	**Years**	**Source**	**Original Range**	**Transformed Range**	**Transformation**
Population Density	2014	WorldPop[1]	(0.113, 116.909)	(-1.543, 4.762)	$\log(x + 0.1)$
Night Time Light	2013	National Centers for Environmental Information[2]	(0.001, 6.300)	(-0.693, 4.151)	$\log(x + 0.5)$
Vegetation Index	2013–2014	NASA EOSDIS Land Processes DAAC[3]	(-1.744, 2.974)	(-1.744, 2.974)	x
Average Temperature	Mean value over the period 1970–2000	WorldClim[4]	(-3.490,1.432)	(-3.490,1.432)	x
Precipitation	Mean value over the period 1970–2000	WorldClim[5]	(0.060, 22.187)	(-1.835, 3.104)	$\log(x + 0.1)$
Access to Nearest City	2000	Joint Research Centre of the European Commission[6]	(0.001, 5.633)	(-8.862, 1.728)	$\log x$

[1] http://www.worldpop.org.uk/data/get_data/. Unit is number of people per pixel.

[2] https://ngdc.noaa.gov/eog/dmsp/downloadV4composites.html.

[3] https://lpdaac.usgs.gov/dataset_discovery/modis/modis_products_table/mod13a3_v006.

[4] http://worldclim.org/version2. Unit is in Celsius temperature (recentered).

[5] http://worldclim.org/version2. Unit: in millimeters.

[6] http://forobs.jrc.ec.europa.eu/products/gam/download.php. Measured in travel time.

model [49] the weighted estimate is modeled and random effects are introduced. In a unit-level model, individual level data are modeled [11], and this is the path we follow. [130] provides a comprehensive overview of SAE. For the Kenya data, Figure 2.1 gives Admin 1, weighted estimates along with the widths of 90% (asymptotic) confidence intervals (based on a variance estimate that accounts for the complex design). In the left panel, we see higher levels of WHZ in the west and south, with lower, less desirable, outcomes in the north and east. In the right panel, we see that the accompanying uncertainty measures are wide, so that the weighted estimates are producing imprecise estimates , especially within the areas in the northwest region, making it difficult to interpret the weighted estimates.. We aim to reduce uncertainty using spatial smoothing and covariate modeling. We emphasize that we model at the point level, but the final deliverable for policy making is at the area-level, so that we are required to aggregate from points to areas.

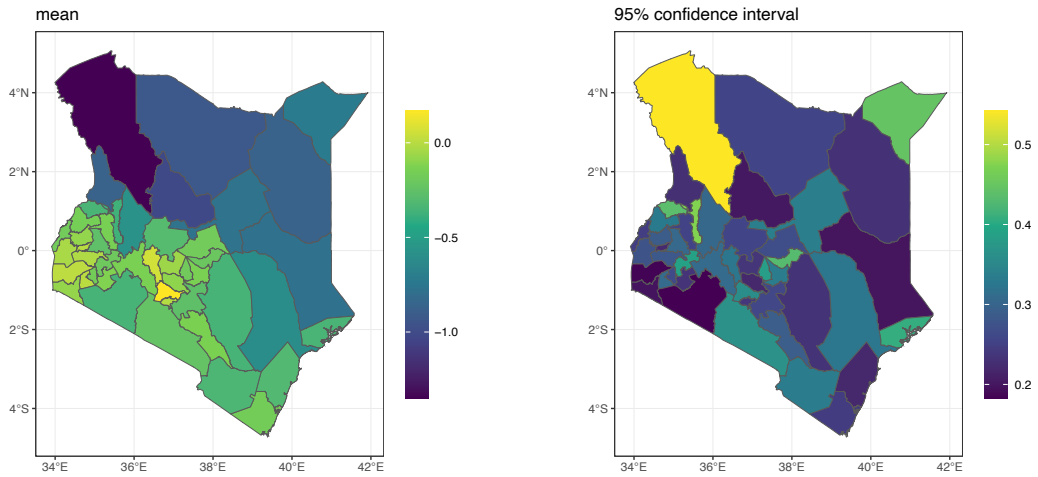

Figure 2.1: Maps of Admin 1 level WHZ weighted estimates (left) and 90% confidence interval widths (right).

2.3 Model Description

To describe the data framework and introduce notation, we will outline a conventional traditional spatial model with a linear model in Section 2.3.1. We will then introduce BART in Section 2.3.2 and propose our spatial version in Section 2.3.3. We describe the

priors in Section 2.3.4.

2.3.1 Model formulation and preliminaries

We consider the spatial dataset $(\mathbf{Y}, \boldsymbol{x})$, where $\mathbf{Y}$ is the collection of observed outcomes and $\boldsymbol{x}$ is the covariate matrix. Let n be the number of spatial locations observed (in our example this corresponds to clusters), noting that we will allow multiple outcomes (these are children in our case) to be observed at the same location so that $\boldsymbol{y} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_n)^\intercal$ is a jagged array of vectors, with n_i being the number of observations in cluster i and $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{in_i})^\intercal$ being the vector of all observations at the same location. Specifically, we let $y_{ik} := y_k(\boldsymbol{s}_i)$ be the k-th response observed at (two-dimensional) location $\boldsymbol{s}_i$, $i = 1, \ldots, n$. We let P be the number of covariates (including the intercept), so that $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iP})$ is the vector of covariates observed at location $\boldsymbol{s}_i$ where x_{ip} is the value of covariate p observed at cluster i for $p = 1, \ldots, P$ (for simplicity of notation, we assume that observations at the same spatial unit share the same covariate values). We assume a Gaussian error model for the observations with measurement error variance σ_e^2. The assumption of constant variance across locations is important, and we should endeavor to check this. We let the latent random spatial component at location $\boldsymbol{s}_i$ be denoted $z(\boldsymbol{s}_i)$, $i = 1, \ldots, n$, and assume that is follow a GRF with Matérn covariance function over the region.

A simple linear mixed effects model with a spatial random field can be written as:

$$y_{ik} \mid \boldsymbol{x}_i, \boldsymbol{\beta}, \boldsymbol{z}, \sigma_e^2 \overset{i.i.d}{\sim} N\left(\boldsymbol{x}_i^\top \boldsymbol{\beta} + z_i, \sigma_e^2\right), \tag{2.1}$$

$$\boldsymbol{z} \mid \boldsymbol{\psi} \sim GF(0, \sigma_m^2 \boldsymbol{\Sigma}), \tag{2.2}$$

$k = 1, \ldots, n_i$, $i = 1, \ldots, n$. The $n \times n$ matrix $\boldsymbol{\Sigma}$ is a Matérn correlation matrix and σ_m^2 is the spatial variance. The correlation matrix has elements,

$$\Sigma_{ij} = \mathrm{Corr}_M\left(z\left(\boldsymbol{s}_i\right), z\left(\boldsymbol{s}_j\right)\right) = \frac{2^{1-\upsilon}}{\Gamma(\upsilon)}\left(\kappa \left\|\boldsymbol{s}_i - \boldsymbol{s}_j\right\|\right)^\nu K_\nu\left(\kappa \left\|\boldsymbol{s}_i - \boldsymbol{s}_j\right\|\right), \tag{2.3}$$

for $i, j = 1, \ldots, n$, $i \neq j$, and with κ being the scale parameter of the GRF and $K_\upsilon(\cdot)$ is the modified Bessel function. The smoothness parameter υ is usually fixed *a priori*, because data sparsity makes it difficult to estimate. For the rest of Chapter 2, we set the value of υ to the default value used in [93]. There is a one-to-one relationship between the scale

parameter κ and the range parameter (which is the more commonly interpreted parameter) with $\rho = \sqrt{8\upsilon}/\kappa$. We let $\boldsymbol{\psi} = (\sigma_m^2, \rho)$ be the set of Matérn covariance parameters. The model is completed with priors on the regression parameters $\boldsymbol{\beta}$ and a prior on the variance components,

$$\pi(\sigma_e^2, \sigma_m^2, \kappa). \tag{2.4}$$

Details on the particular prior we use are postponed until Section 2.3.4.

For latent Gaussian models (LGMs), a closed-form expression for $\pi(\boldsymbol{z}, \boldsymbol{\beta}, \sigma_e^2, \sigma_m^2, \kappa \mid \boldsymbol{y})$ is computationally expensive for large n, due to the computation of the inverse and determinant of the dense $n \times n$ covariance matrix. Efficient MCMC sampling algorithms are difficult to construct for spatial models because of the strong dependence between parameters [86]. [142] showed that the INLA method, originally an approximation Bayesian inference procedure based on Laplacian approximations and numerical integration rules, is available for LGMs. The INLA approach provides approximations for the marginal posterior distributions of the latent Gaussian random field and the hyperparameters, along with a reliable estimate for the marginal distribution $\pi(\boldsymbol{y})$ [74].

For models with continuously indexed GRFs, [93] showed that a discrete Gaussian Markov random field (GMRF) can be used to approximate the continuously indexed data GRF generated through a Matérn covariance function, via a stochastic partial differential equation (SPDE) approach. This not only reduces the computation complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^{3/2})$, but also enables model inference to be carried out through INLA.

2.3.2 The BART model

The Bayesian Additive Regression Trees (BART) model [30] can be viewed as the summation of a fixed number of highly flexible nonparametric regression functions, where each function $g(\cdot; T_j, \boldsymbol{\mu}_j) : \mathbb{R}^p \to \mathbb{R},\ j = 1, \ldots, m$ is a random function that maps the P-dimensional covariate space onto the real line, with a total number of $l = 1, \ldots, m$ functions. The component functions are defined by two sets of parameters, the binary tree structure $\mathbf{T} = (T_1, \ldots, T_m)$ and the terminal node values $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m)$. Each binary tree can be viewed as a set of decision rules that splits the covariate space into a finite number of regions and at

any internal node, for a given tree, a 'splitting variable' is chosen from the P covariates and a threshold value is chosen that splits the current region into two sub-regions. A fitted value is then assigned to the corresponding region at each end node, according to the terminal node values $\boldsymbol{\mu}$. By splitting according to different covariate variables and threshold values, BART is capable of modeling complex nonlinear relationships and interactions among covariates. Furthermore, a regularization prior that penalizes tree complexity in order to avoid model overfitting is assigned to the tree structure parameters so that the trees functions are 'weak learners' [30].

2.3.3 Sampling Model and Latent Field

The sampling model for BARTSIMP is a combination of the linear mixed effects model described in Section 2.3.1 with the BART terms of Section 2.3.2 replacing the linear model. Substituting the linear covariate effects with the sum-of-trees model gives:

$$y_{ik} \mid z_i, \boldsymbol{x}_{ip}, \mathbf{T}, \boldsymbol{\mu}, \sigma_e^2 \overset{i.i.d}{\sim} N\left(\sum_{l=1}^{m} g\left(\boldsymbol{x}_{ip}; T_l, \boldsymbol{\mu}_l\right) + z_i, \sigma_e^2\right)$$

with the distribution of $\boldsymbol{z}$ and Matérn covariance being as previously defined in Equations (2.2) and (2.3).

2.3.4 The Priors

We assume a priori independence between the tree parameters $(\mathbf{T}, \boldsymbol{\mu})$, the residual variance parameter σ_e^2 and the spatial hyperparameters $\boldsymbol{\psi}$:

$$p\left(\mathbf{T}, \boldsymbol{\mu}, \sigma_e^2, \boldsymbol{\psi}\right) = \left[\prod_{j=1}^{m} p\left(\boldsymbol{\mu}_j \mid T_j\right) p\left(T_j\right)\right] p\left(\sigma_e^2\right) p\left(\boldsymbol{\psi}\right).$$

For the tree parameters and the residual variance parameter, we follow the prior specifications in [29] and decompose the tree structure prior into three hierarchical parts: the probability of a node being non-terminal, the probability of choosing one of the covariates as the splitting variable for a non-terminal node and the probability of choosing a splitting value given the chosen splitting variable. For a tree node at depth d, the probability of it being a non-terminal node is given the form $\alpha(1+d)^{-\beta}$, with $\alpha \in (0,1), \beta \in [0,\infty)$. In the

examples In this chapter, we defer to the default settings in [30] and choose $\alpha = 0.95$ and $\beta = 2$. Further, we assume that the splitting variable is uniformly chosen among all covariates for each internal node, and the splitting value is uniformly chosen from all distinct values of the selected splitting variable. We assume independent and identically distributed normal priors for the terminal node values $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m$ given their corresponding tree structure T_j. The mean and variance for the normal prior are chosen such that each tree will only function as a 'weak learner' that contributes a small part in the model; we refer to [30] for further details.

For the residual variance parameter σ_e^2, we take the default choice in [30] and specify a scaled inverse-Gamma distribution $\sigma_e^2 \sim \nu\lambda/\chi_\nu^2$, where ν is the degrees-of-freedom. The values of the hyperparameters (ν, λ) are chosen through an exploratory data analysis procedure to give $\hat{\sigma}_r^2$ as an empirical guess of σ_e^2 through a tentative working model (for example, an SPDE model with all covariates as linear predictors), which matches the q-th quantile of the scaled inverse Gamma prior. Following [30], we let $(q, \nu) = (0.9, 3)$. Hence, the prior depends on the data in a weak fashion.

Finally, we place a penalized complexity (PC) prior on the Matérn hyperparameters $\boldsymbol{\psi}$, following [54]. The joint PC prior for $\boldsymbol{\psi}$ is

$$p(\boldsymbol{\psi}) = \frac{d}{2}\tilde{\lambda}_1\tilde{\lambda}_2\rho^{-d/2-1}\exp\left(-\tilde{\lambda}_1\rho^{-d/2} - \tilde{\lambda}_2\sigma_m\right), \sigma_m > 0, \rho > 0,$$

where $d = 2$, $\tilde{\lambda}_1 = -\log(\alpha_1)\rho_0^{d/2}$ and $\tilde{\lambda}_2 = -\log(\alpha_2)/\sigma_0$. The hyperparameters $(\alpha_1, \alpha_2, \rho_0, \sigma_0)$ guarantee that $\mathrm{P}(\rho < \rho_0) = \alpha_1$ and $\mathrm{P}(\sigma_m > \sigma_0) = \alpha_2$. We let $\alpha_1 = \alpha_2 = 0.5$ and set (ρ_0, σ_m) to the crude estimates obtained from fitting a model with only an intercept and the spatial random field. Hence, we again specify a data dependent prior.

2.4 Computation

2.4.1 Overview on the Algorithm

For the standard BART model, an efficient Gibbs-type sampler is available under conjugate priors on the model parameters [29]. However, the computation faces scalability challenges once the spatial random effect component is added into the model. Under our model framework, a standard MCMC algorithm requires the sampling and storage of the

spatial random field $\boldsymbol{z}$, which is computationally undesirable. Hence, our strategy is to integrate out $\boldsymbol{z}$ and operate on the corresponding marginal likelihood. However, the calculation of the marginal likelihood involves computing the determinant of large and dense variance-covariance matrices, which becomes prohibitive as the number of datapoints at distinct locations (clusters in the Kenya example) becomes large. As a result, we consider the 'INLA-within-MCMC' technique proposed by [61]. Specifically, the algorithm uses INLA to approximate the aforementioned marginal likelihood, which is then used to compute the Metropolis-Hastings acceptance ratio in the MCMC routine for the remaining model parameters.

We conclude this section with a brief overview of the INLA-within-MCMC method, and refer the readers to Section 2.4.2 for full details of the computation algorithm for BARTSIMP. We let $\boldsymbol{\theta}$ denote the whole ensemble of hyperparameters and latent effects. Further, we consider the following partition $\boldsymbol{\theta} = (\boldsymbol{\theta}_c, \boldsymbol{\theta}_{-c})$, where $\boldsymbol{\theta}_{-c}$ represents the set of parameters we would like to integrate out, and $\boldsymbol{\theta}_c$ being the set of parameters for which we will use a Metropolis-Hastings algorithm. The posterior distribution can be re-expressed as

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{y}) \propto \pi\left(\boldsymbol{y} \mid \boldsymbol{\theta}_{-c}, \boldsymbol{\theta}_c\right) \pi\left(\boldsymbol{\theta}_{-c} \mid \boldsymbol{\theta}_c\right) \pi\left(\boldsymbol{\theta}_c\right).$$

Integrating $\boldsymbol{\theta}_{-c}$ from both sides gives,

$$\pi\left(\boldsymbol{\theta}_c \mid y\right) \propto \pi\left(\boldsymbol{y} \mid \boldsymbol{\theta}_c\right) \pi\left(\boldsymbol{\theta}_c\right). \tag{2.5}$$

In the context of the Metropolis-Hastings algorithm, let $\boldsymbol{\theta}_c^{\star}$ be the potential value for the new iteration proposed by the transition kernel $q\left(\cdot \mid \boldsymbol{\theta_c}\right)$. A new $\boldsymbol{\theta}_c$ is proposed within each iteration of the MCMC routine. Plugging (2.5) into the expression of the acceptance probability, we have,

$$\alpha = \min\left\{1, \frac{\pi\left(\boldsymbol{y} \mid \boldsymbol{\theta}_c^*\right) \pi\left(\boldsymbol{\theta}_c^*\right) q\left(\boldsymbol{\theta}_c \mid \boldsymbol{\theta}_c^*\right)}{\pi\left(\boldsymbol{y} \mid \boldsymbol{\theta}_c\right) \pi\left(\boldsymbol{\theta}_c\right) q\left(\boldsymbol{\theta}_c^* \mid \boldsymbol{\theta}_c\right)}\right\}. \tag{2.6}$$

With the prior distribution $\pi(\cdot)$ and proposal distribution $q(\cdot \mid \cdot)$ for $\boldsymbol{\theta}_c$ specified, the computation bottleneck in (2.6) is to calculate $\pi\left(\boldsymbol{y} \mid \boldsymbol{\theta}_c\right)$ and $\pi\left(\boldsymbol{y} \mid \boldsymbol{\theta}_c^{\star}\right)$. In practice, it can be computationally expensive to obtain an exact likelihood. The INLA-within-MCMC approach provides an efficient method to provide approximations to $\pi\left(\boldsymbol{y} \mid \boldsymbol{\theta}_c\right)$ and $\pi\left(\boldsymbol{y} \mid \boldsymbol{\theta}_c^{\star}\right)$, by fitting models with R-INLA, with the values of $\boldsymbol{\theta_c}$ (or $\boldsymbol{\theta_c}^{\star}$) fixed.

2.4.2 Algorithmic Details on BARTSIMP

We next provide an overview of the Metropolis-within-Gibbs algorithm we will be using for the BARTSIMP model, where the INLA-within-MCMC approach is applied to approximate the acceptance ratio. We let $T_{-j}, \boldsymbol{\mu}_{-j}$ be the set of trees and terminal node values, respectively, not including T_j and $\boldsymbol{\mu}_j$. An outline of the MCMC routine is given in Algorithm 1. We now describe each of the updates.

Algorithm 1: An overview of the BARTSIMP algorithm.

Input: Initialized values for $\mathbf{T}, \boldsymbol{\mu}, \sigma_e^2, \psi$ for a number of MCMC iterations B.

for $b \leftarrow 1$ **to** B **do**

  for $j \leftarrow 1$ **to** m **do**

    Update $T_j \mid \boldsymbol{y}, T_{-j}, \boldsymbol{\mu}_{-j}, \sigma_e^2, \boldsymbol{\psi}$.

    Update $\boldsymbol{\mu}_j \mid \boldsymbol{y}, T_j, T_{-j}, \boldsymbol{\mu}_{-j}, \sigma_e^2, \boldsymbol{\psi}$.

    Update $\sigma_e^2 \mid \boldsymbol{y}, \mathbf{T}, \boldsymbol{\mu}, \boldsymbol{\psi}$.

    Update $\boldsymbol{\psi} \mid \boldsymbol{y}, \mathbf{T}, \boldsymbol{\mu}, \sigma_e^2$.

  end

end

Update $T_j \mid \boldsymbol{y}, T_{-j}, \boldsymbol{\mu}_{-j}, \sigma_e^2, \psi$

To update the sum-of-trees variables, we follow the Bayesian backfitting MCMC algorithm that is outlined in Section 3.1 of [29]. Specifically, we carry out a sequential update of the set of tree structure parameters and terminal node values $\{(T_1, \boldsymbol{\mu}_1), \ldots, (T_m, \boldsymbol{\mu}_m)\}$, updating each tree, one at a time. To update each of the tree structure parameters T_j, we integrate out $\boldsymbol{\mu}_j$ and do a Metropolis-within-Gibbs update of T_j, while keeping the other parameters, $(T_{-j}, \boldsymbol{\mu}_{-j}, \sigma_e^2, \boldsymbol{\psi})$, fixed. The acceptance ratio for T_j is,

$$\alpha_{T_j} = \min \left\{ 1, \frac{\pi\left(\boldsymbol{y} \mid T_j^*, T_{-j}, \boldsymbol{\mu}_{-j}, \sigma_e^2, \boldsymbol{\psi}\right) \pi\left(T_j^*\right) q\left(T_j \mid T_j^*\right)}{\pi\left(\boldsymbol{y} \mid T_j, T_{-j}, \boldsymbol{\mu}_{-j}, \sigma_e^2, \boldsymbol{\psi}\right) \pi\left(T_j\right) q\left(T_j^* \mid T_j\right)} \right\}. \tag{2.7}$$

Here, T_j^* is the proposed structure for tree j, following the proposal distribution for T_j in [30], through which we can compute the transition kernel ratio $q\left(T_j \mid T_j^*\right)/q\left(T_j^* \mid T_j\right)$. To compute the likelihood ratio, we employ the backfitting algorithm as follows. For the likelihood $\pi\left(\boldsymbol{y} \mid T_j, T_{-j}, \boldsymbol{\mu}_{-j}, \sigma_e^2, \boldsymbol{\psi}\right)$, we see that, given the tree parameters of all trees except for T_j, we can compute the residual values with respect to the remaining $m-1$ trees. Denote these vectors of residuals as $\boldsymbol{r}_1^{(j)}, \ldots, \boldsymbol{r}_n^{(j)}$, where

$$r_{ik}^{(j)} = y_{ik} - \sum_{l \neq j} g\left(\boldsymbol{x}_i; T_l, \boldsymbol{\mu}_l\right), \qquad l = 1, \ldots, m, \quad i = 1, \ldots, n, \quad k = 1, \ldots, n_i.$$

We see that the original model is now equivalent to a single-treed model (along with the spatial field and the Gaussian noise) with response $(\boldsymbol{r}_1^{(j)}, \ldots, \boldsymbol{r}_n^{(j)})$, where the superscript (j) is with respect to the particular tree that is being 'left out'. Since the tree structure for T_j is conditioned upon, the tree part is equivalent to a linear model $\mathbf{C}^{(j)}\boldsymbol{\mu}_j$ where $\boldsymbol{\mu}_j = (\mu_{j1}, ..., \mu_{jb_j})$ is the set of terminal node parameters in $\boldsymbol{\mu}_j$, and $\mathbf{C}^{(j)}$ is the $n \times b_j$ covariate matrix with elements,

$$C_{it}^{(j)} = 1 \text{ if } \boldsymbol{x}_{[i]} \text{ belongs to terminal node } t \text{ in tree } T_j, \text{ otherwise } C_{it}^{(j)} = 0,$$

where $C_{it}^{(j)}$ is the element in the i-th row and t-th column of $\mathbf{C}^{(j)}$. Thus, given $T_j, T_{-j}, \boldsymbol{\mu}_{-j}$, the whole model is equivalent to:

$$\begin{aligned} \boldsymbol{r}^{(j)} \mid \mathbf{C}^{(j)}, \boldsymbol{\mu}_j, \sigma_e^2, \boldsymbol{\psi} &\sim N\left(\mathbf{C}^{(j)}\boldsymbol{\mu}_j, \sigma_e^2\mathbf{I}_n + \boldsymbol{\Sigma}(\boldsymbol{\psi})\right) \\ \boldsymbol{\mu}_j &\sim N(\mathbf{0}, \kappa\mathbf{I}_{b_j}). \end{aligned} \tag{2.8}$$

Therefore, the likelihood term $\pi\left(\boldsymbol{y} \mid T_j, T_{-j}, \boldsymbol{\mu}_{-j}, \sigma_e^2, \boldsymbol{\psi}\right)$ in (2.7) can be expressed as

$$\begin{aligned} \pi\left(\boldsymbol{y} \mid T_j, T_{-j}, \boldsymbol{\mu}_{-j}, \sigma_e^2, \boldsymbol{\psi}\right) &= \pi(\boldsymbol{r}^{(j)} \mid \mathbf{C}^{(j)}(\boldsymbol{x}), \sigma_e^2, \boldsymbol{\psi}) \\ &= \mathcal{N}(\mathbf{0}; \mathbf{0}, \kappa\mathbf{C}^{(j)}\mathbf{C}^{(j)\intercal} + \boldsymbol{\Sigma} + \sigma_e^2\mathbf{I}_n), \end{aligned} \tag{2.9}$$

where $\mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the likelihood for a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, evaluated at $\boldsymbol{y}$. On examination of (2.9), we see that a major computational challenge for generating posterior samples for $\boldsymbol{\mu}_j$ arises from computing the inverse of $\left(\kappa\mathbf{C}^{(j)}\mathbf{C}^{(j)\intercal} + \boldsymbol{\Sigma} + \sigma_e^2\mathbf{I}_n\right)$, which has complexity $\mathcal{O}(n^3)$. To reduce the computational

burden, we first re-express the first line in (2.8) as a hierarchical model, with the spatial effect $\boldsymbol{z}$ being conditioned upon:

$$\begin{aligned} \boldsymbol{r}^{(j)} \mid \mathbf{C}^{(j)}, \boldsymbol{z}, \sigma_e^2 &\overset{i.i.d}{\sim} N\left(\mathbf{C}^{(j)}\boldsymbol{\mu}_j + \boldsymbol{z}, \sigma_e^2\mathbf{I}_d\right) \\ \boldsymbol{z} \mid \boldsymbol{\psi} &\sim GF(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\psi})). \end{aligned} \tag{2.10}$$

The GRF in (2.8) can be approximated as a GMRF [93]:

$$\begin{aligned} \boldsymbol{r}^{(j)} \mid \mathbf{C}^{(j)}, \mathbf{A}, \boldsymbol{u}, \sigma_e^2 &\overset{i.i.d}{\sim} N\left(\mathbf{C}^{(j)}\boldsymbol{\mu}_j + \mathbf{A}\boldsymbol{u}, \sigma_e^2\mathbf{I}_d\right) \\ \boldsymbol{u} &\sim N(\mathbf{0}, \mathbf{Q}^{-1}) \end{aligned} \tag{2.11}$$

where $\boldsymbol{u}$ is the GMRF, $\mathbf{A}$ is a projection matrix that projects the random effects on mesh points to the spatial locations, and $\mathbf{Q}$ is the sparse precision matrix for $\boldsymbol{u}$. Thus we can approximate the likelihood term in (2.9) as $\mathcal{N}\left(\mathbf{0}; \mathbf{0}, \kappa\mathbf{C}^{(j)}{\mathbf{C}^{(j)}}^{\top} + \mathbf{A}\mathbf{Q}^{-1}\mathbf{A}^{\intercal} + \sigma_e^2\mathbf{I}_n\right)$. The computation for $\pi\left(\boldsymbol{y} \mid T_j^*, T_{-j}, \boldsymbol{\mu}_{-j}, \sigma_e^2, \boldsymbol{\psi}\right)$ is similar.

Update $\boldsymbol{\mu}_j \mid \boldsymbol{y}, T_j, T_{-j}, \boldsymbol{\mu}_{-j}, \sigma_e^2, \psi$

The model formulation in (2.8) leads to the posterior distribution for $\boldsymbol{\mu}_j$:

$$\boldsymbol{\mu}_j \mid \mathbf{C}^{(j)}, \boldsymbol{r}^{(j)}, \sigma_e^2, \boldsymbol{\psi}, \kappa \sim N(\boldsymbol{m}_j, \mathbf{V}_j)$$

where

$$\begin{aligned} \mathbf{V}_j &= \left[\mathbf{C}^{(j)\intercal}\left(\sigma_e^2\mathbf{I}_n + \boldsymbol{\Sigma}\right)^{-1}\mathbf{C}^{(j)} + \kappa^{-1}\mathbf{I}_{b_j}\right]^{-1} \\ \boldsymbol{m}_j &= \mathbf{V}_j^{-1}\mathbf{C}^{(j)\intercal}\left(\sigma_e^2\mathbf{I}_n + \boldsymbol{\Sigma}\right)^{-1}\boldsymbol{r}^{(j)}. \end{aligned} \tag{2.12}$$

Here, the major computation burden in (2.12) is to compute the inverse of $\left(\sigma_e^2\mathbf{I}_n + \boldsymbol{\Sigma}\right)$. Similar to Section 2.4.2, we can approximate the GP with GMRF and expresse the approximated posterior distribution for $\boldsymbol{\mu}_j$ as

$$\boldsymbol{\mu}_j \mid \mathbf{C}^{(j)}, \boldsymbol{r}^{(j)}, \sigma_e^2, \boldsymbol{\psi}, \kappa \sim N(\tilde{\boldsymbol{m}}_j, \tilde{\mathbf{V}}_j)$$

where

$$\begin{aligned} \tilde{\mathbf{V}}_j &= \left[\mathbf{C}^{(j)\intercal}\left(\sigma_e^2\mathbf{I}_n + \mathbf{A}\mathbf{Q}^{-1}\mathbf{A}\right)^{-1}\mathbf{C}^{(j)} + \kappa^{-1}\mathbf{I}_{b_j}\right]^{-1} \\ \tilde{\boldsymbol{m}}_j &= \mathbf{V}_j^{-1}\mathbf{C}^{(j)\intercal}\left(\sigma_e^2\mathbf{I}_n + \mathbf{A}\mathbf{Q}^{-1}\mathbf{A}\right)^{-1}\boldsymbol{r}^{(j)}. \end{aligned} \tag{2.13}$$

To further simplify the computation, note that the most computationally costly steps in computing (2.13) are inverting $\mathbf{Q}$ and $\sigma_e^2\mathbf{I}_n + \mathbf{A}\mathbf{Q}^{-1}\mathbf{A}$. By the Woodbury matrix identity we have,

$$\left(\sigma_e^2\mathbf{I}_n + \mathbf{A}\mathbf{Q}^{-1}\mathbf{A}\right)^{-1} = \sigma_e^{-2}\mathbf{I} - \sigma_e^{-4}\mathbf{A}(\mathbf{Q} + \sigma_e^{-2}\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{A}^\intercal. \tag{2.14}$$

Note that compared to (2.13), in this expression, we do not need to explicity invert $\mathbf{Q}$, and also note that $\mathbf{Q} + \sigma_e^{-2}\mathbf{A}^\intercal\mathbf{A}$ is a sparse matrix, which leads to much lower computational cost.

Update $\sigma_e^2 \mid \boldsymbol{y}, \mathbf{T}, \boldsymbol{\mu}, \boldsymbol{\psi}$ *and* $\boldsymbol{\psi} \mid \boldsymbol{y}, \mathbf{T}, \boldsymbol{\mu}, \sigma_e^2$

As with the tree structure parameters $(T_1, \ldots, T_m)$, we can use the Metropolis-within-Gibbs method to update the residual variance parameter σ_e^2 and use the INLA-within-MCMC technique to approximate the Metropolis-Hastings acceptance ratio:

$$\alpha_{\sigma_e^2} = \min\left\{1, \frac{\pi\left(\boldsymbol{y} \mid \mathbf{T}, \boldsymbol{\mu}, \psi, \sigma_e^{2*}\right)\pi\left(\sigma_e^{2*}\right)q\left(\sigma_e^2 \mid \sigma_e^{2*}\right)}{\pi\left(\boldsymbol{y} \mid \mathbf{T}, \boldsymbol{\mu}, \psi, \sigma_e^2\right)\pi\left(\sigma_e^2\right)q\left(\sigma_e^{2*} \mid \sigma_e^2\right)}\right\}.$$

To ensure that the variance parameter is positive, we use a Gaussian proposal for $\log\sigma_e^2$. We again use the backfitting technique to compute the overall residual $\boldsymbol{r}$, defined as all m tree terms subtracted from the response value:

$$r_{ik} = y_{ik} - \sum_{l=1}^{m} g\left(\boldsymbol{x}_i; \mathbf{T}_l, \boldsymbol{\mu}_l\right), \qquad k = 1, \ldots, n_i, \quad i = 1, \ldots, n.$$

This is equivalent to the following sampling model:

$$\begin{aligned}\boldsymbol{r} \mid \boldsymbol{z}, \sigma_e^2 &\overset{i.i.d}{\sim} N\left(\boldsymbol{z}, \sigma_e^2\right)\\ \boldsymbol{z} \mid \boldsymbol{\psi} &\sim GF(\mathbf{0}, \boldsymbol{\Sigma}).\end{aligned}$$

By fitting a linear mixed effect model with residual variance fixed at σ_e^2 and spatial hyperparameters fixed at $\boldsymbol{\psi}$, we can compute the approximated marginal likelihood of the model $\pi(\boldsymbol{r} \mid \sigma_e^2, \boldsymbol{\psi})$, which is equivalent to $\pi\left(\boldsymbol{y} \mid \mathbf{T}, \boldsymbol{\mu}, \boldsymbol{\psi}, \sigma_e^2\right)$. We update the spatial hyperparameters in a similar fashion.

2.4.3 R and C++ Integration

We implemented the backbone of the MCMC algorithm for BARTSIMP based on the C++ code in the `BART` package [146]. In order to use the functions in the `R-INLA` package [104] to carry out the INLA computation during the Metropolis-Hastings adjustment step, we used Rcpp [47, 45, 46] as an interface between C++ and R. To provide an open-source computing software for BARTSIMP, we developed the R package `BARTSIMP`, with source code available at `https://github.com/AlexJiang1125/BARTSIMP`.

2.4.4 Comparison of Computation Time: Details on the BARTSIMP-exact

Consider a simple multivariate Gaussian model with zero mean and Matern covariance matrix:

$$\boldsymbol{y} \mid \sigma_e^2, \sigma_m^2, \kappa \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma} + \sigma_e^2 \mathbf{I}),$$

where

$$\Sigma_{ij} = \sigma_m^2 \frac{2^{1-\nu}}{\Gamma(\nu)} (\kappa d_{ij})^{\nu} K_{\nu} (\kappa d_{ij}).$$

here, we let $\nu = 1, \sigma_e^2 = \sigma_m^2 = 4, \kappa = 2$. Let n be the number of observations in $\boldsymbol{y}$, the marginal likelihood has form

$$-\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det(\boldsymbol{\Sigma} + \sigma_\varepsilon^2 \mathbf{I}) - \frac{1}{2}\boldsymbol{y}^T(\boldsymbol{\Sigma} + \sigma_\varepsilon^2 \mathbf{I})^{-1}\boldsymbol{y}. \tag{2.15}$$

The method 'exact' will evaluate (2.15) directly. The formula can also be approximate using SPDE where the dense covariance matrix is approximated with the inverse of a sparse precision matrix, (this is implemented by INLA). Table shows the averages and standard deviations (over 10 replications) of algorithm running times for both two methods, under values n 100, 500, 1000 and 5000:

	$n = 100$	$n = 500$	$n = 1000$	$n = 5000$
INLA-SPDE (BARTSIMP)	2.897	3.223	3.356	3.453
	(0.334)	(0.201)	(0.164)	(0.523)
Exact (BARTSIMP-exact)	0.022	0.201	1.725	183.9
	(0.004)	(0.025)	(0.199)	(1.980)

Table 2.2: Comparison between INLA-SPDE and the exact method under different number of observations.

2.5 Simulation Experiments

2.5.1 Simulation setting

In this section, we study several simulation scenarios in which the spatial and covariate signals have different strengths. We consider a 50×50 grid surface over a study region $[0, 1] \times [0, 1]$ and denote the set of grid cells as G. For the covariates associated with each grid cell, we independently generate two covariates from a uniform distribution on $[0, 1]$. Let $x_{gp}, p = 1, 2$, represents the p-th covariate for grid cell g and let $\boldsymbol{x}_g = (x_{g1}, x_{g2}), g \in G$. The true deterministic field evaluated at cell g, denoted $f(\boldsymbol{x}_g)$, is defined as:

$$f(\boldsymbol{x}_g) = (1 - \omega)\boldsymbol{z}^\star_g + \omega f_0(\boldsymbol{x}_g), \qquad g \in G,$$

where the 'baseline' spatial field $\boldsymbol{z}^\star$ is generated from a GRF with Matérn parameters $\kappa = 2.5$ (note that this corresponds to $\rho = \frac{\sqrt{8v}}{\kappa} \approx 1.13$), $\sigma^2_m = 0.5$. We let $f_0(\boldsymbol{x}_g)$ be the 'baseline' covariate surface. Depending on the structure in the covariate surface, we have three different scenarios for $f_0(\boldsymbol{x}_g)$:

- **Tree-structured**: $f_0(\boldsymbol{x}_g)$ is generated based on the following step function:

$$f_0(\boldsymbol{x}_g) = \begin{cases} 3 & x_{g1} < 0.5 \\ 0 & x_{g1} \geq 0.5, x_{g2} \geq 0.5 \\ -2 & x_{g1} \geq 0.5, x_{g2} < 0.5 \end{cases} \cdot$$

- **Linear**: $f_0(\boldsymbol{x}_g)$ is a linear combination of x_{g1} and x_{g2}:

$$f_0(\boldsymbol{x}_g) = 2x_{g1} - x_{g2}.$$

- **Smooth**: $f_0(\boldsymbol{x}_g)$ is a smooth function of x_{g1} and x_{g2}:

$$f_0(\boldsymbol{x}_g) = \sin(2\pi x_{g1}) + \cos(2\pi x_{g2}).$$

The scalar ω is fixed at different values and can be interpreted as the proportion of 'covariate signal' among the overall signal. Here we consider five different scenarios for ω: 1 (covariate signal only), 0.8 (strong covariate signal), 0.5 (medium covariate signal), 0.2 (weak covariate signal) and 0 (spatial signal only), which we denote as scenarios 1, 2, 3, 4 and 5. We then randomly select 250 cells from the grid and randomly sample one location uniformly within each cell, ending up with 250 spatial points over the study region, mimicking the spatial locations for the clusters. The number of observations for each spatial location is sampled from a uniform distribution over $\{5, 6, 7, 8, 9, 10\}$. For each observation at a given spatial location, its value is defined as the sum of the deterministic field value (spatial field plus covariate signal) from the grid cell it belongs to, and Gaussian random noise with $\sigma_e^2 = 1$. We simulate 10 datasets for each scenario. Note that the covariate and spatial surface are independently generated for each dataset, and not held constant. For the BARTSIMP and BART model, we used ensembles of 20 trees, and collected 2,000 posterior samples after a burn-in period of 2,000. We set the model hyperparameters for the PC prior choices $(\alpha_1, \alpha_2, \rho_0, \sigma_m)$ and scaled inverse-Gamma prior choices (ν, q) to the default values described in Section 2.3.4. We also conducted a sensitivity analyses in 2.6.1 to show that the model estimation results are robust to the choice of hyperparameter values.

We compare BARTSIMP against several alternatives, which can be grouped into two categories depending on whether a BART component is included in the model. For the BART-related methods, we consider a standard BART model without a spatial latent field **(BART)** and a modified version of BARTSIMP, in which the likelihood function is calculated using the exact formula, instead of the SPDE approximation **(BARTSIMP-EXACT)**. We include this approach to assess whether the INLA approximation to the

marginal distribution is accurate. For the non-BART methods, we consider a GMRF spatial model with Matérn covariance function fitted by the SPDE approach, including all covariates in a linear model **(SPDE)** and a spatial SPDE model, with intercept only, i.e., no covariates, **(SPDE0)**. Additionally, we implemented an alternative version of the SPDE model, with a first-order random walk processes on the covariates **(SPDE-RW1)**.

2.5.2 Performance Criteria

We examine the models via performance measures that evaluate both point and interval estimates, over the gridded surface:

- **Root mean squared error (RMSE)**: Let $\hat{f}^{(j)}(\boldsymbol{x}_g)$ be the model prediction for $f(\boldsymbol{x}_g)$ given by method j, the RMSE is defined as:

$$\text{RMSE}^{(j)} = \frac{1}{|G|}\sum_{g\in G}\left(\hat{f}^{(j)}(\boldsymbol{x}_g) - f(\boldsymbol{x}_g)\right)^2, \qquad j = 1,\ldots,5.$$

- **Average interval length (AIL)**: Let $(L^{(j)}_{g,\alpha}, U^{(j)}_{g,\alpha})$ denote the $100\times(1-\alpha)\%$ prediction interval for $f(\boldsymbol{x}_g)$, available from method j. The AIL is defined as the average length of prediction intervals over the gridded surface:

$$\text{AIL}^{(j)} = \frac{1}{|G|}\sum_{g\in G}|U^{(j)}_{g,\alpha} - L^{(j)}_{g,\alpha}|, \qquad j = 1,\ldots,5$$

- **Average coverage rate (ACR)**: The ACR is defined as the coverage of $f(\boldsymbol{x}_g)$, for the interval, $(L^{(j)}_{g,\alpha}, U^{(j)}_{g,\alpha})$:

$$\text{ACR}^{(j)} = \frac{1}{|G|}\sum_{g\in G}\text{I}\left(f(\boldsymbol{x}_g) \in L^{(j)}_{g,\alpha}, U^{(j)}_{g,\alpha})\right), \qquad j = 1,\ldots,5.$$

  For AIL and ACR we take $\alpha = 0.05$.

- **Average Interval score (AIS)**: The AIS is an integrated metric for prediction interval accuracy defined as:

$$\text{AIS}^{(j)} = \frac{1}{|G|}\sum_{g\in G}\Bigg[(U^{(j)}_{g,\alpha} - L^{(j)}_{g,\alpha}) + \frac{2}{\alpha}\left(L^{(j)}_{g,\alpha} - f(\boldsymbol{x}_g)\right)\cdot\mathbf{1}(f(\boldsymbol{x}_g) < L^{(j)}_{g,\alpha}) + \frac{2}{\alpha}\left(f(\boldsymbol{x}_g) - U^{(j)}_{g,\alpha}\right)\cdot\mathbf{1}(f(\boldsymbol{x}_g) > U^{(j)}_{g,\alpha})\Bigg], \qquad j = 1,\ldots,5.$$

The AIS can be broken down into three parts, with the first part penalizing wider intervals and the second and third parts penalizing low coverage rates [59].

Among the measures, RMSE assesses the point prediction accuracy, while AIL, ACR and AIS focus on the uncertainty prediction accuracy.

2.5.3 Results

Figures 2.2 shows comparisons of RMSE, AIL, ACR and AIS across all six methods under the five scenarios, with the covariate surface generated from a tree-structured model. First, we observe that BARTSIMP and BARTSIMP-EXACT perform similarly for all four metrics under all scenarios and mechanisms, suggesting that the approximation error from SPDE is negligible in model fitting. For point prediction accuracy, BART (the covariate-only model) has the lowest RMSE when there is only covariate signal, and both SPDE and SPDE0 (the spatial models) have the lowest RMSE when there is only spatial signal. Meanwhile, BARTSIMP (and BARTSIMP-EXACT) perform the best across all methods for the remaining scenarios where the true data is generated from a mixture of covariate and spatial signals. For uncertainty estimation, BARTSIMP performs best out of all methods in all scenarios, having coverage rates closest to the nominal coverage. As spatial signals become stronger, SPDE and SPDE0 tend to have lower interval widths and interval scores, and both BART and BARTSIMP tend to have wider interval widths and interval scores.

BARTSIMP and BARTSIMP-EXACT perform similarly, but their computation time differs. While the exact method is quicker for data with few observations, the approximation method is considerably more efficient as the number of observations grows. Attaining the nominal coverage is an issue for all approaches, particular for the stronger spatial signal cases, but we think this is just the reality of modeling point-level spatial data without large sample sizes. The same behavior of under-coverage was seen in the comprehensive examination pf [125] (Figure 2.1 in this Chapter, in particular, clearly shows this phenomenon).

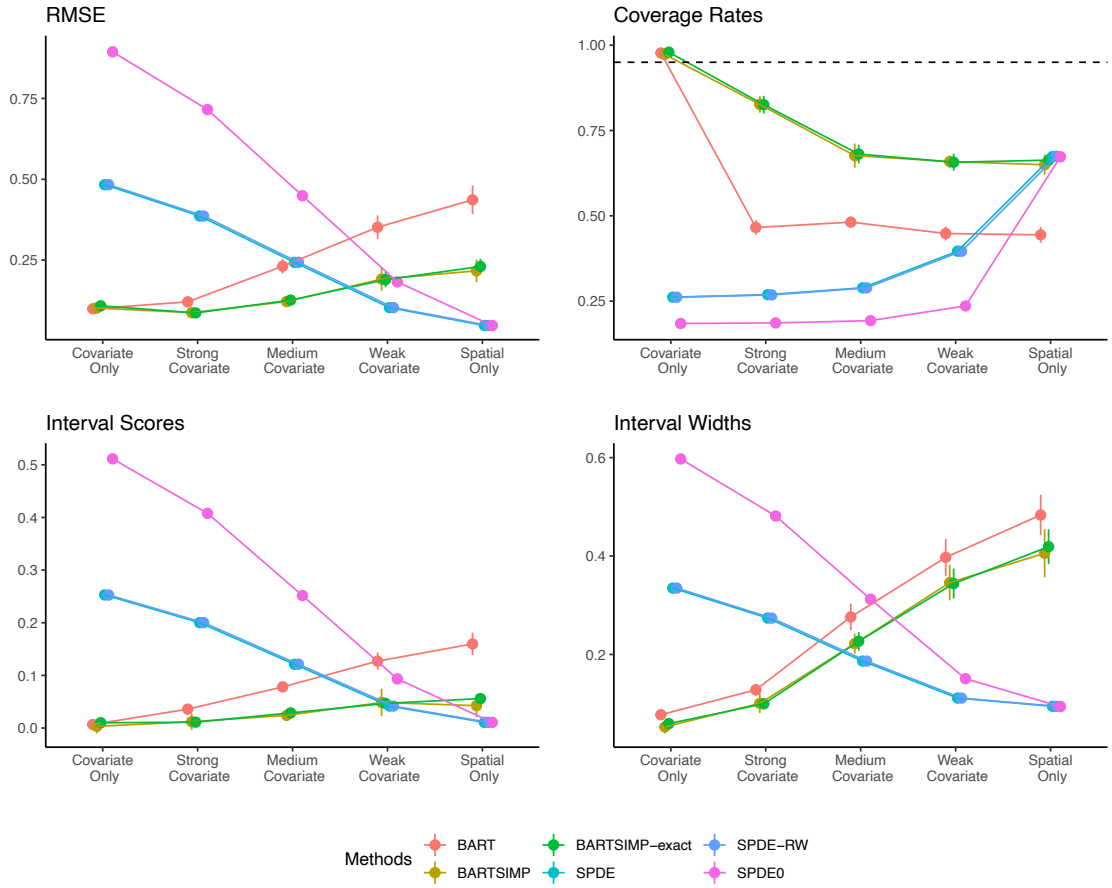

Figure 2.2: Root mean squared error (RMSE), average coverage rate (ACR), average interval length (AIL) and average interval score (AIS) for five different methods under five scenarios (covariate only, strong, medium, weak covariate signals and spatial signals only), when the covariate surface has a **tree-based structure**. The dots show the average values over 10 replications, and the vertical lines illustrate the 95% Wald type interval computed from the standard deviation over 10 replications. The horizontal dashed line in the upper right figure illustrates the 95% nominal coverage. The dashed horizontal line represents the nominal coverage rate (0.95). The horizontal positions of the dots and lines are slightly jittered to avoid overlapping, as some methods have very similar metrics.

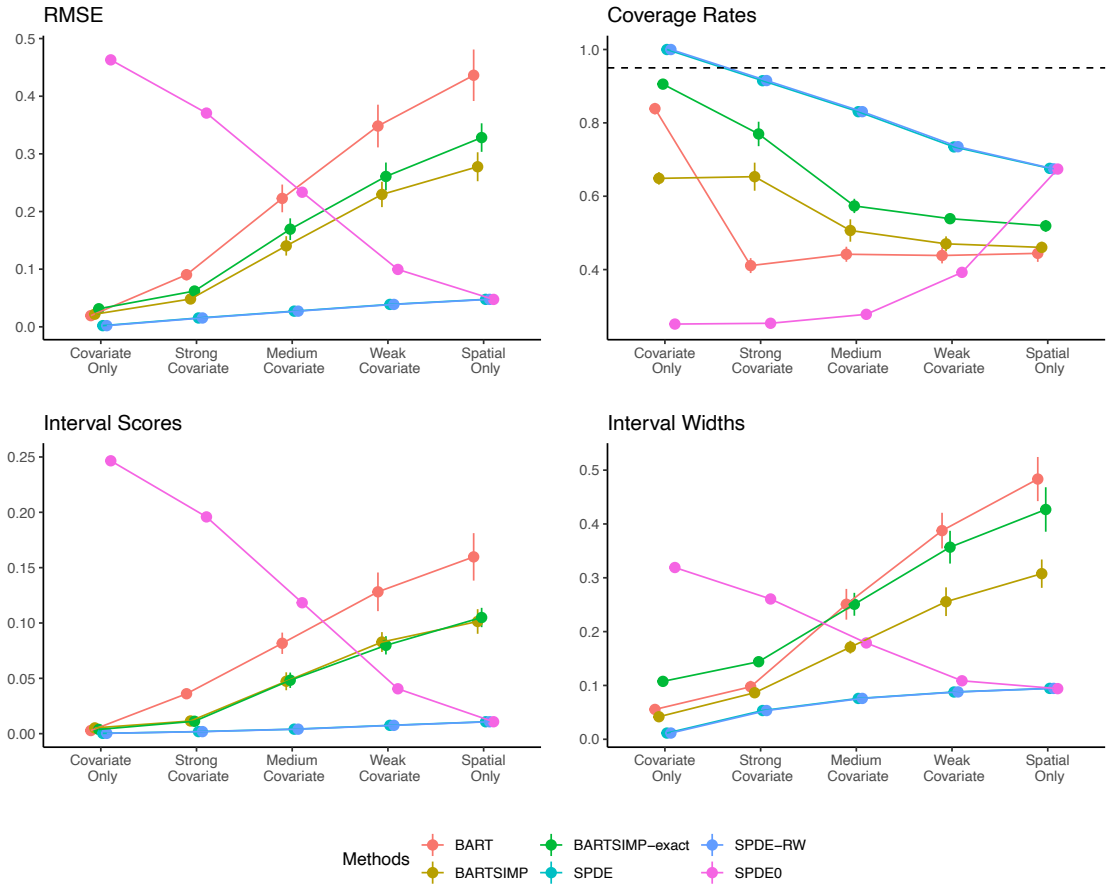

Figure 2.3: Root mean squared error (RMSE), average coverage rate (ACR), average interval length (AIL) and average interval score (AIS) for five different methods under five scenarios (covariate only, strong, medium, weak covariate signals and spatial signals only), when the covariate surface has a **linear structure**. The dots show the average values over 10 replications, and the vertical lines illustrate the 95% Wald type interval computed from the standard deviation over 10 replications. The horizontal dashed line in the upper right figure illustrates the 95% nominal coverage. The dashed horizontal line represents the nominal coverage rate (0.95). The horizontal positions of the dots and lines are slightly jittered to avoid overlapping, as some methods have very similar metrics.

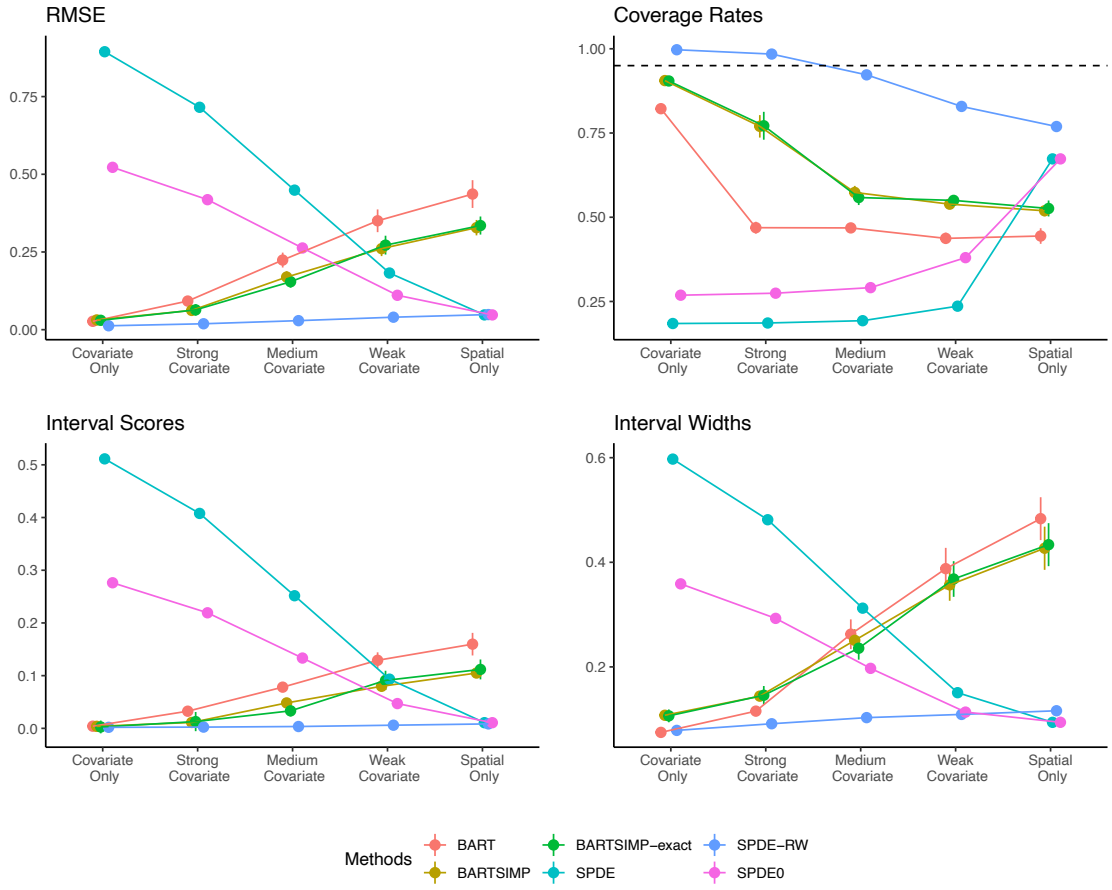

Figure 2.4: Root mean squared error (RMSE), average coverage rate (ACR), average interval length (AIL) and average interval score (AIS) for five different methods under five scenarios (covariate only, strong, medium, weak covariate signals and spatial signals only), when the covariate surface has a **smooth structure**. The dots show the average values over 10 replications, and the vertical lines illustrate the 95% Wald type interval computed from the standard deviation over 10 replications. The dashed horizontal line represents the nominal coverage rate (0.95). The horizontal dashed line in the upper right figure illustrates the 95% nominal coverage.

The results for cases where the covariate surfaces are generated from a linear (or smooth) model are shown in Figure 2.3 and 2.4. We note that the BART, BARTSIMP and BARTSIMP-

EXACT still perform in a very similar fashion to as they did in the tree-structured case, where better prediction performances are associated with stronger covariate signals. For the case where the covariate surface is generated from a linear model, SPDE and SPDE-RW consistently outperform other methods in terms of most prediction metrics. This is not surprising, as the linear covariate effect can be correctly specified by these methods (however, note that all three BART-related methods have very similar RMSE when only the covariate signal is present in the data). As the model does not include covariate effects, SPDE0 performs worse than other methods when the covariate signal is strong, but has better performance as the spatial signal increases.

When the covariate surface is generated from a smooth model, only SPDE-RW outperforms the other methods, as the smooth function cannot be correctly specified by models including SPDE. Note that BARTSIMP (and BARTSIMP-EXACT) still attain similar prediction performance when the covariate signal is relatively strong.

2.5.4 Convergence Checks

2.6 Posterior Convergence Checks

Finally, we also conducted convergence analyses on the posterior samples from the simulation studies under all three scenarios. We look at the posterior convergence for the following quantities:

- σ_m^2, σ^2 and κ: the spatial hyperparameters
- $\sum_{j=1}^{m} g(\boldsymbol{x}_i; \mathbf{T}_j, \boldsymbol{\mu}_j), i \in \{400, 500, 750\}$: the fitted covariate values for the 400, 500 and 750-th row in the dataset. We refer to them as 'location 1', 'location 2' and 'location 3'.

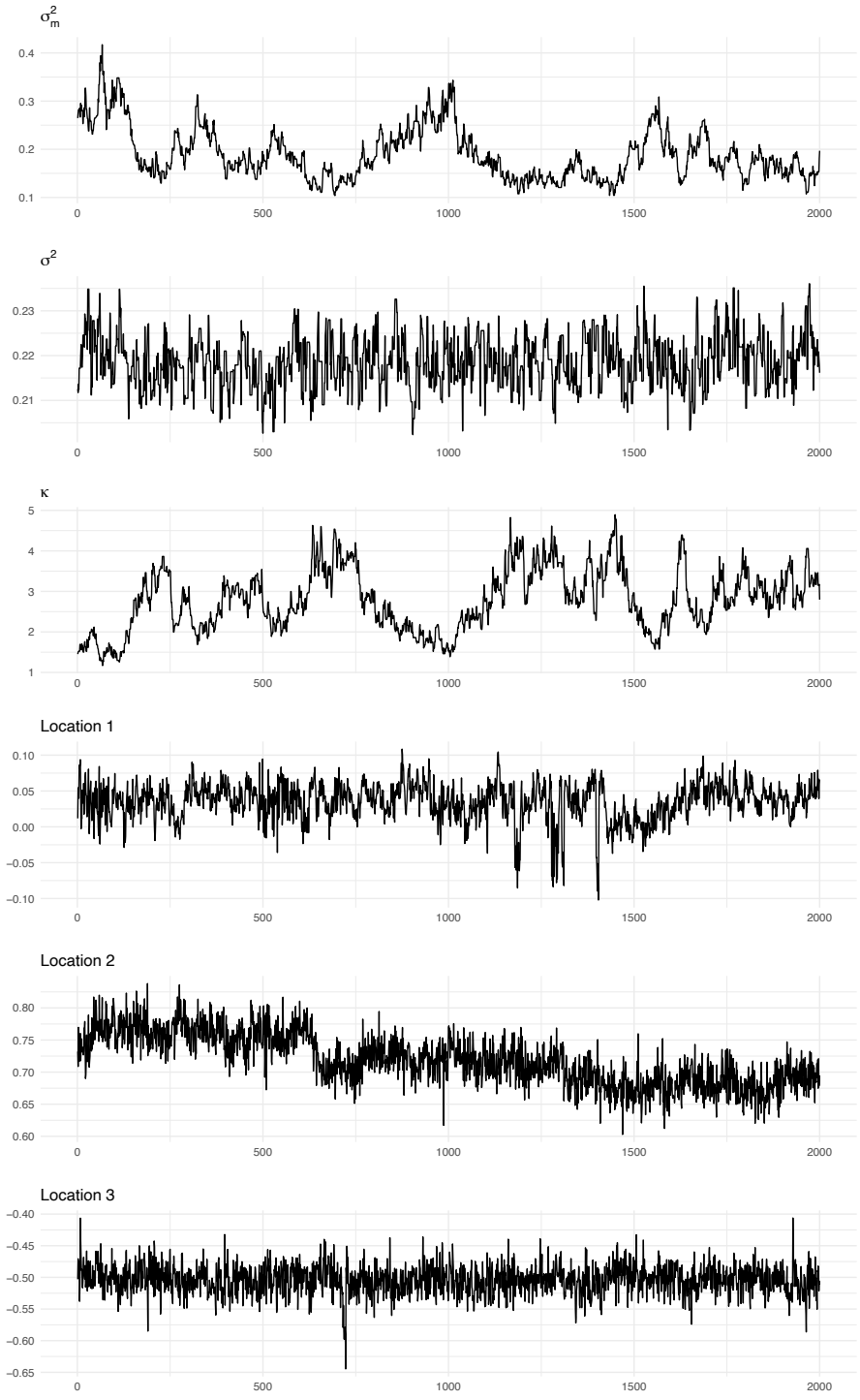

Figure 2.5: Line plots for the six quantites obtained from the 2001-4000 posterior samples in dataset 1.

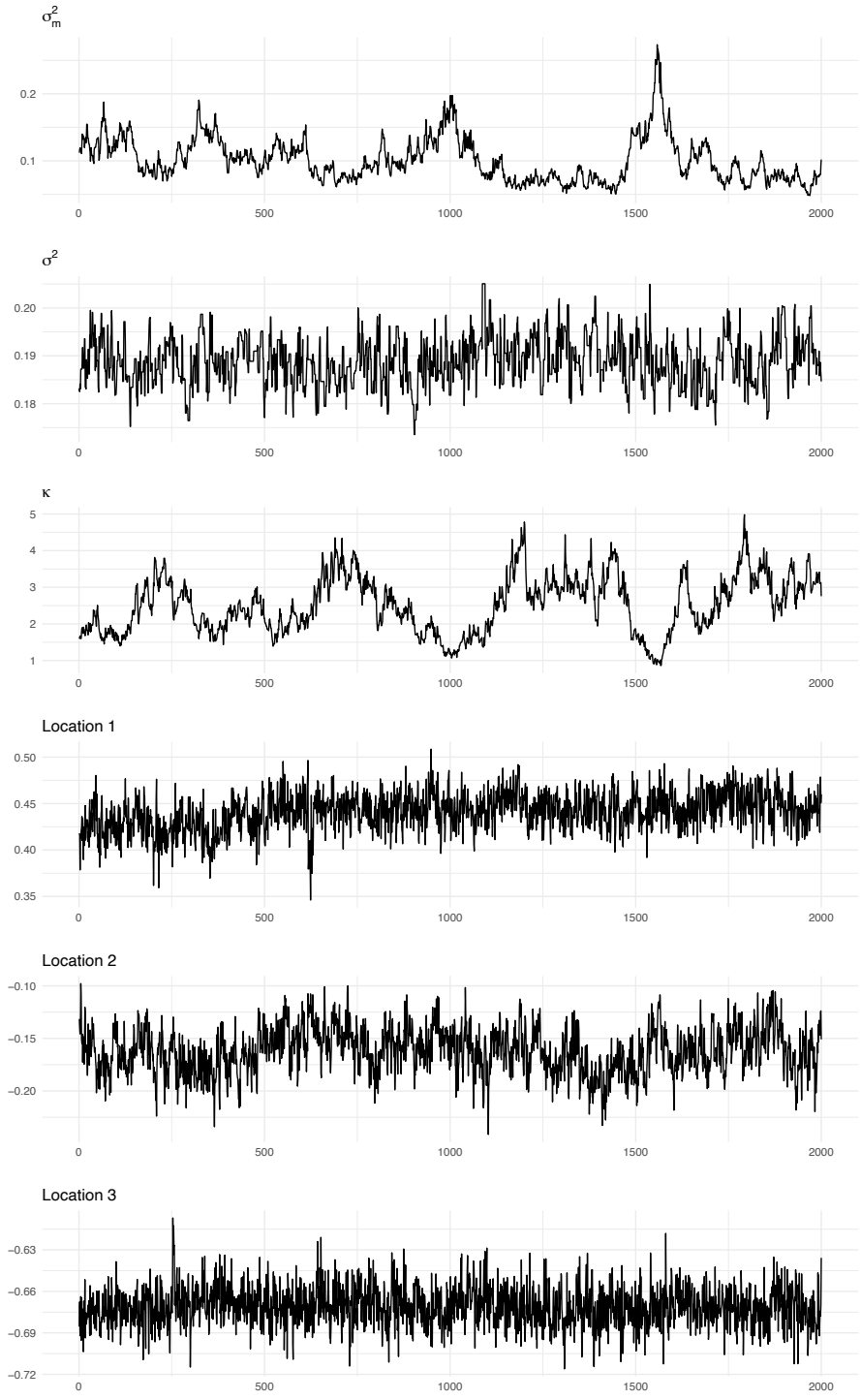

Figure 2.6: Line plots for the six quantites obtained from the 2001-4000 posterior samples in dataset 2.

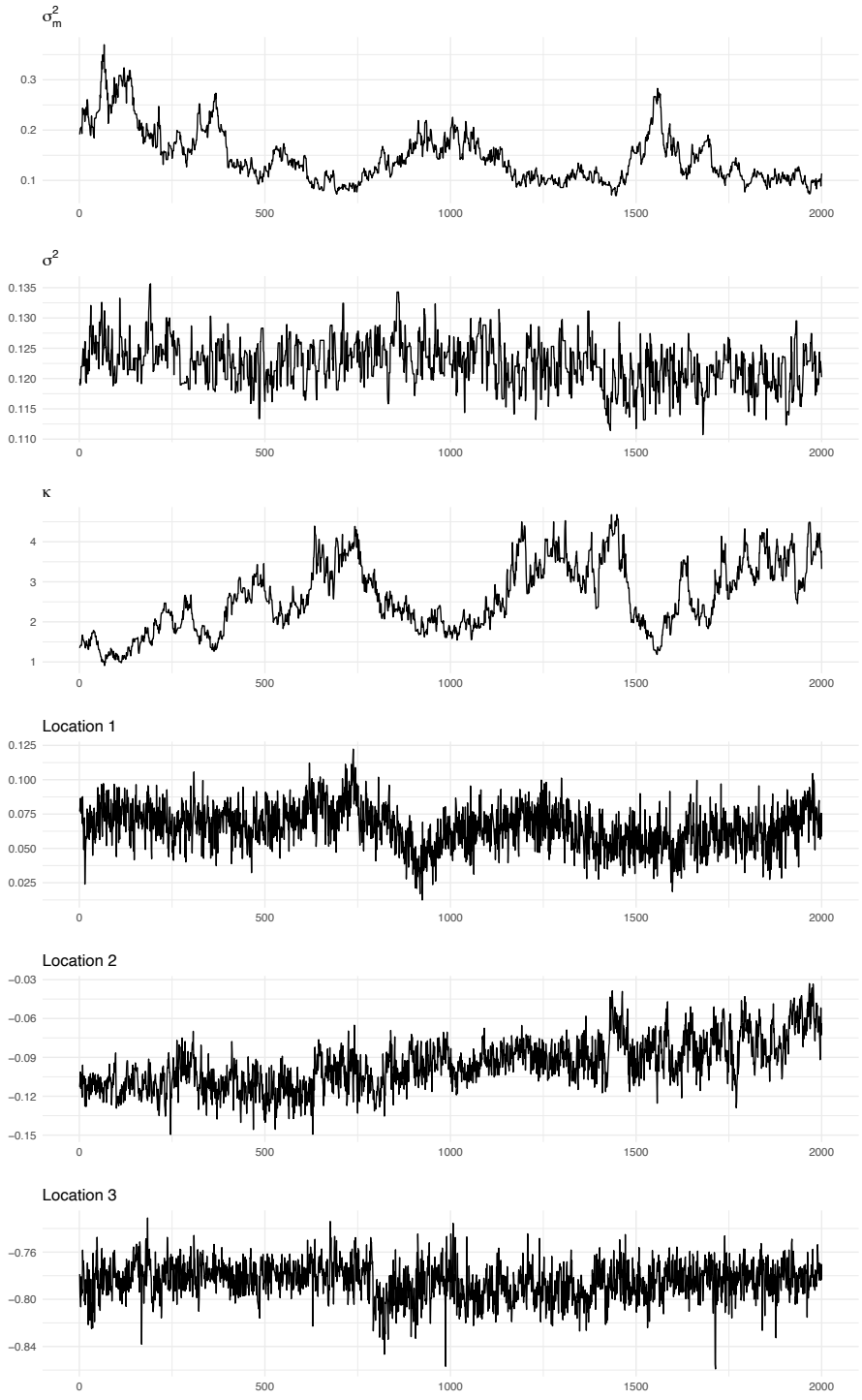

Figure 2.7: Line plots for the six quantites obtained from the 2001-4000 posterior samples in dataset 3.

Figures 2.5 to 2.7 in Section 2.6 show line plots of the 2001st-4000th posterior samples. Table 2.3 shows the efficient sample sizes obtained from the 2000 samples.

2.6.1 Sensitivity analysis on the model hyperparameters

Default choices of the hyperparameters

We use the following setting as the default prior specification. For the hyperparameters in the BART prior, we let $\alpha = 0.95, \beta = 2$ for the tree structure and $\nu = 3, q = 0.95$ for the residual variance parameter, following the default values recommended by [30]. For the PC prior for the Matérn parameters, we let $\alpha_1 = \alpha_2 = 0.5, \rho_0 = 2.4, \sigma_0 = 0.5$. In the following subsections, we conduct a series of sensitivity analyses on a set of model hyperparameters, while keeping the values of the remaining parameters to their default choices.

Here, we recall the prior distributions chosen for the parameters ρ, σ_m and σ_e^2. The residual variance is assigned the scaled inverse-Gamma prior, i.e. $\sigma_e^2 \sim \nu\lambda/\chi_\nu^2$, where the scale parameter λ is chosen based on the quantile q for $\hat{\sigma}_r^2$, which is an 'educated guess' on the residual variance based on a tentative working model (for more detailed information, see Section 3.4 in this Chapter). The spatial hyperparameter σ_m and ρ are assigned the PC-prior distribution [54] as follows:

$$p(\rho, \sigma_m) = \frac{d}{2}\tilde{\lambda}_1\tilde{\lambda}_2\rho^{-d/2-1}\exp\left(-\tilde{\lambda}_1\rho^{-d/2} - \tilde{\lambda}_2\sigma_m\right), \sigma_m > 0, \rho > 0,$$

where $d = 2$, $\tilde{\lambda}_1 = -\log(\alpha_1)\rho_0^{d/2}$ and $\tilde{\lambda}_2 = -\log(\alpha_2)/\sigma_0$. Note that ρ and σ_m are factorizable in the prior, and they are correspondingly controlled by hyperparameters (α_1, ρ_0) and (α_2, σ_m).

An illustration of the priors, under different sets of hyperparameters.

To intuitively show the difference in distributions of ρ, σ_m and σ_e^2 under different choices of hyperparameter values, we plot the probability density functions of these parameters (in the same order) in Figure 2.8. Note that in the case for σ_e^2, we have chosen $\hat{\sigma}_r^2 = 1$ in order to chose a scale parameter for the scaled inverse-Gamma distribution. We see that shape of the probability density function differs significant under different choices of hyperparameters.

scenarios	σ_m^2	σ_e^2	κ	$\sum_{j=1}^{m} g\left(\boldsymbol{x}_{400}; \mathbf{T}_j, \boldsymbol{\mu}_j\right)$	$\sum_{j=1}^{m} g\left(\boldsymbol{x}_{500}; \mathbf{T}_j, \boldsymbol{\mu}_j\right)$	$\sum_{j=1}^{m} g\left(\boldsymbol{x}_{750}; \mathbf{T}_j, \boldsymbol{\mu}_j\right)$
1	4.94	380.26	14.68	476.93	355.51	854.33
2	12.44	262.06	16.79	152.80	63.37	168.93
3	17.74	303.88	22.52	79.34	168.82	176.20
4	20.21	298.31	28.15	54.03	65.46	148.07
5	22.46	332.86	30.10	64.84	156.82	173.77
6	5.07	307.26	14.51	136.44	151.67	414.23
7	12.61	254.22	16.13	100.25	128.84	127.22
8	15.87	305.55	22.00	87.83	100.86	293.35
9	17.01	314.94	25.09	130.24	68.70	128.08
10	22.46	332.86	30.10	64.84	156.82	173.77
11	5.27	204.71	14.90	109.08	101.36	312.14
12	13.04	246.62	16.13	102.05	115.11	365.53
13	16.31	300.75	22.27	79.89	111.96	69.95
14	20.78	324.56	26.43	100.54	73.72	157.05
15	22.46	332.86	30.10	64.84	156.82	173.77

Table 2.3: Efficient sample sizes of the quantities of interest, averaged over 10 datasets for each scenario. Each row represents a different scenario (1–5 for 'tree structured', 6–10 for 'linear', and 11–15 for 'smooth').

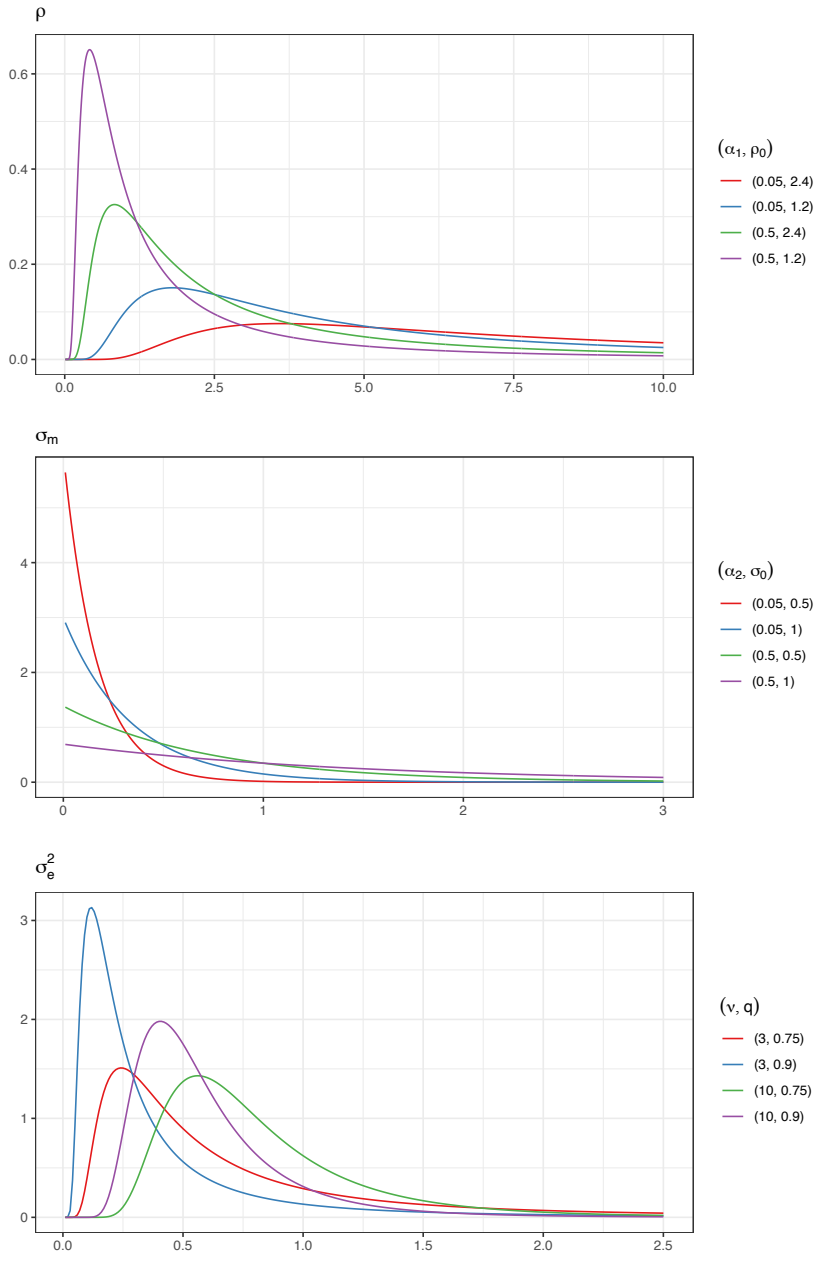

Figure 2.8: Probability density functions of ρ, σ_m and σ_e^2 under different values of hyperparameters. The different color labels represent different choices of hyperparameters.

Sensitivity analysis of α_1 and α_2

We consider the following four scenarios of α_1 and α_2: (1) $\alpha_1 = 0.5, \alpha_2 = 0.5$, (2) $\alpha_1 = 0.05, \alpha_2 = 0.5$, (3) $\alpha_1 = 0.5, \alpha_2 = 0.05$, (4) $\alpha_1 = 0.05, \alpha_2 = 0.05$. Note that scenario (1) corresponds to the default values of α_1 and α_2.

Hyperparameter Settings	RMSE	AIS	AIL	ACR
$\alpha_1 = 0.05, \alpha_2 = 0.05$	0.0971	0.0148	0.1585	82.78%
	(0.0129)	(0.0046)	(0.0292)	(0.0692)
$\alpha_1 = 0.05, \alpha_2 = 0.5$	0.0973	0.0149	0.1584	82.54%
	(0.0129)	(0.0046)	(0.0289)	(0.0693)
$\alpha_1 = 0.5, \alpha_2 = 0.05$	0.0975	0.0149	0.1584	82.50%
	(0.0130)	(0.0046)	(0.0291)	(0.0692)
$\alpha_1 = 0.5, \alpha_2 = 0.5$	0.0971	0.0147	0.1587	82.78%
	(0.0128)	(0.0044)	(0.0292)	(0.0685)

Table 2.4: Prediction performance measures over all four scenarios of α_1 and α_2 combinations. The nominal coverage is 95% and small values of AIS are preferred. The averages over 25 test datasets are shown with standard deviations shown in brackets.

Sensitivity analysis of ρ_0 and σ_0

We consider the following four scenarios of ρ_0 and σ_0: (1) $\rho_0 = 2.4, \sigma_0 = 0.5$, (2) $\rho_0 = 2.4, \alpha_2 = 1$, (3) $\rho_0 = 1.2, \sigma_0 = 0.5$, (4) $\rho_0 = 1.2, \sigma_0 = 1$. Note that scenario (1) corresponds to the default values of ρ_0 and σ_0.

Hyperparameter Settings	RMSE	AIS	AIL	ACR
$\rho_0 = 2.4, \sigma_0 = 0.5$	0.0963	0.0146	0.1580	82.10%
	(0.0127)	(0.0041)	(0.0255)	(0.0580)
$\rho_0 = 2.4, \sigma_0 = 1$	0.0963	0.0146	0.1579	82.27%
	(0.0127)	(0.0041)	(0.0253)	(0.0596)
$\rho_0 = 1.2, \sigma_0 = 0.5$	0.0962	0.0145	0.1580	82.35%
	(0.0127)	(0.0041)	(0.0252)	(0.0594)
$\rho_0 = 1.2, \sigma_0 = 1$	0.0961	0.0144	0.1582	82.61%
	(0.0126)	(0.0040)	(0.0256)	(0.0582)

Table 2.5: Prediction performance measures over all four scenarios of ρ_0 and σ_0 combinations. The nominal coverage is 95% and small values of AIS are preferred. The averages over 25 test datasets are shown with standard deviations shown in brackets.

Sensitivity analysis of ν and q

We consider the following four scenarios of ν and q: (1) $\nu = 3, q = 0.75$, (2) $\nu = 3, q = 0.9$, (3) $\nu = 10, q = 0.75$, (4) $\nu = 10, q = 0.9$. Note that scenario (1) corresponds to the default values of ν and q.

Hyperparameter Settings	RMSE	AIS	AIL	ACR
$\nu = 3, q = 0.75$	0.0988	0.0152	0.1590	82.21%
	(0.0135)	(0.0044)	(0.0248)	(0.0615)
$\nu = 3, q = 0.9$	0.0961	0.0145	0.1580	82.48%
	(0.0127)	(0.0041)	(0.0252)	(0.0589)
$\nu = 10, q = 0.75$	0.0967	0.0146	0.1597	83.15%
	(0.0129)	(0.0039)	(0.0232)	(0.0523)
$\nu = 10, q = 0.9$	0.0964	0.0147	0.1577	82.23%
	(0.0148)	(0.0044)	(0.0236)	(0.0581)

Table 2.6: Prediction performance measures over all four scenarios of ν and q combinations. The nominal coverage is 95% and small values of AIS are preferred. The averages over 25 test datasets are shown with standard deviations shown in brackets.

2.6.2 Computation Time

Methods	Average (min)	Standard Deviation (min)
BART	13.7	0.4
BARTSIMP	956.4	261.7
BARTSIMP-exact	608.2	314.6
SPDE	4.9	7.1
SPDE0	5.9	9.6
SPDE-RW	6.4	10.3

Table 2.7: Comparison of the computation time in the simulation study (with average and standard deviation over 10 datasets in 5 scenarios) among the six different methods. The numbers are shown in minutes.

2.7 Application

In this section, we investigate the prediction performance of BARTSIMP, as compared to BART, SPDE and SPDE0, evaluated on the WHZ measurements from the 2014 Kenya DHS.

2.7.1 Cross-Validation Exercise

To compare the prediction performances across different approacahes, we apply cross-validation and split the 1584 clusters in the 2014 DHS survey into training and test datasets of cluster sizes 1267 and 317, respectively. To preserve the stratification structure of the design in both data sets, we consider stratified sampling of the clusters such that the strata proportions in the training set roughly matches that in the test set. We repeated the procedure 10 times to reduce sampling variation caused by using a single data split. We again use RMSE, ACR, AIL and AIS as performance criteria for all four methods, similar to Section 2.5.2, and we use the same settings as before for the BART and BARTSIMP models. The algorithm took one day to run for BARTSIMP on Ubuntu 18.04, with 50GB of memory.

Table 2.8 shows ACR (nominal coverage is 95%), RMSE, AIL and AIS for all approaches. The results are the average over 10 test datasets, with standard deviations in brackets. We see that BARTSIMP and BART dominate the other two methods in terms of having closest to nominal coverage. The BARTSIMP model has the lowest AIS. The SPDE and SPDE0 models both have narrow intervals, but these produce low coverage, which results in poor AIS scores also.

Table 2.8: Prediction performance measures over all four competing methods. The nominal coverage is 95% and small values of AIS are preferred. The averages over 10 test datasets are shown with standard deviations shown in brackets.

	Model	ACR	RMSE	AIL	AIS
1	BART-SIMP	**98**% (0.003)	0.584 (0.024)	6.882 (0.082)	**0.172** (0.002)
2	BART	**93**% (0.005)	0.590 (0.024)	5.287 (0.028)	0.182 (0.005)
3	SPDE	78% (0.010)	**0.565** (0.023)	**3.136** (0.030)	0.256 (0.009)
4	SPDE0	78% (0.008)	0.578 (0.025)	3.156 (0.032)	0.257 (0.010)

2.7.2 Point and Areal Prediction

In this section, we conduct spatial predictions on a grid surface over the study region and generate aggregated estimates at the Admin 1 and Admin 2 levels. The reason we produce estimates at the areal-level is that resource allocation and policy making decisions are made at the area-level.

At location $\boldsymbol{s}$, we let WHZ($\boldsymbol{s}$) be the spatial surface of the height-for-weight Z-scores and $d_5(\boldsymbol{s})$ be the under-five population density. The areal level WHZ is a weighted average over the under-five population density, as the Z-scores were evaluated for children under age five. The $d_5(\boldsymbol{s})$ values are obtained from WorldPop (`https://www.worldpop.org`). The WHZ

for an administrative region R_i is

$$\text{WHZ}_{R_i} = \frac{\int_{R_i} \text{WHZ}(\boldsymbol{s}) d_5(\boldsymbol{s})}{\int_{R_i} d_5(\boldsymbol{s})}, \qquad i = 1, 2, \ldots, m, \tag{2.16}$$

where m is the number of administrative areas.

We approximate the integrals in (2.16) by a weighted sum over observations on grid cells located at $\boldsymbol{s}_g$ over the regions, $g \in R_i$. Let $\text{WHZ}(\boldsymbol{s}_g)$ be the height-for-weight Z-score and $d_5(\boldsymbol{s}_g)$ be the under-five population density evaluated at grid cell g. The regional WHZ, approximated on the grid, is

$$\text{WHZ}_{R_i} \approx \frac{\sum_{g \in R_i} \text{WHZ}(\boldsymbol{s}_g) d_5(\boldsymbol{s}_g)}{\sum_{g \in R_i} d_5(\boldsymbol{s}_g)}. \tag{2.17}$$

Using (2.17), we can calculate the posterior mean and 95% credible interval quantiles for regional WHZ at all Admin 1 and Admin 2 areas, based on the four methods. As a comparison, we also consider the direct weighted areal-level estimates and 95% (design-based) confidence intervals. Figure 2.9 shows the posterior median and 95% credible/confidence intervals, derived from all five methods, and Figure 2.10 shows the predicted areal-level WHZ for all 47 Admin 1 areas in Kenya. The predicted posterior mean for WHZ given by BARTSIMP ranges from -1.10 to 0.05 over Admin 1 regions, showing that there is large within country variation in WHZ in Kenya. Among the 47 Admin 1 regions, Kiambu and Nairobi have the highest WHZ predictions – these are the most populated counties in Kenya. Low WHZ scores occurred in areas such as Turkana, Mandera and Marsabit and these could be targeted for interventions.

We see from Figures 2.9 and 2.10 that while all five methods give quantitatively similar overall patterns of the areal estimate across the regions, BARTSIMP and the spatial methods provide similar point estimates at the local level. BARTSIMP gives relatively wider credible interval lengths, which is consistent with the simulations, in which we saw that these intervals are more appropriate. Finally, BART does not yield reliable point estimates, and gives interval estimates that are far too narrow. We also provide areal-level predictions on Admin 2 levels in Section 2.7.3 (results shown in Figures 2.11 to 2.18). Examination of these results show that the direct estimates for Admin 2 regions have more unreliable point estimates with much wider confidence intervals, due to insufficient samples observed in each

region.

Additionally, an interesting question is whether the GRF models can address the correlation within household. While it is theoretically straightforward to include an additional random household effect to account for dependence on observations in the same household, when we have attempted this in other projects with DHS data, there were identifiability issues, since it was hard to disentangle household and cluster effects.

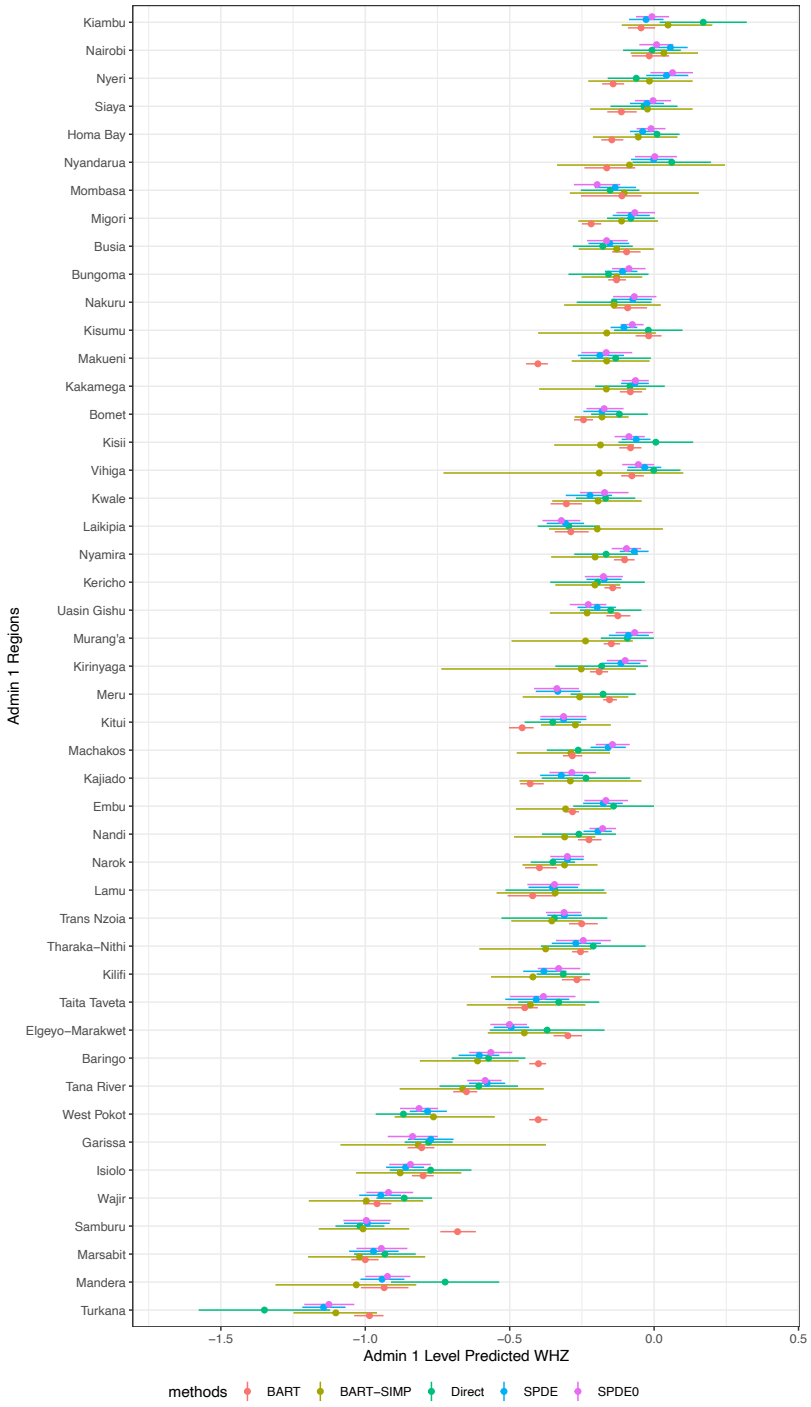

Figure 2.9: The Admin 1 level posterior median and 95% credible/confidence intervals of WHZ for BART, BARTSIMP, SPDE0, SPDE and direct (weighted) estimates. The Admin 1 areas are arranged according to the predicted posterior mean given by BARTSIMP.

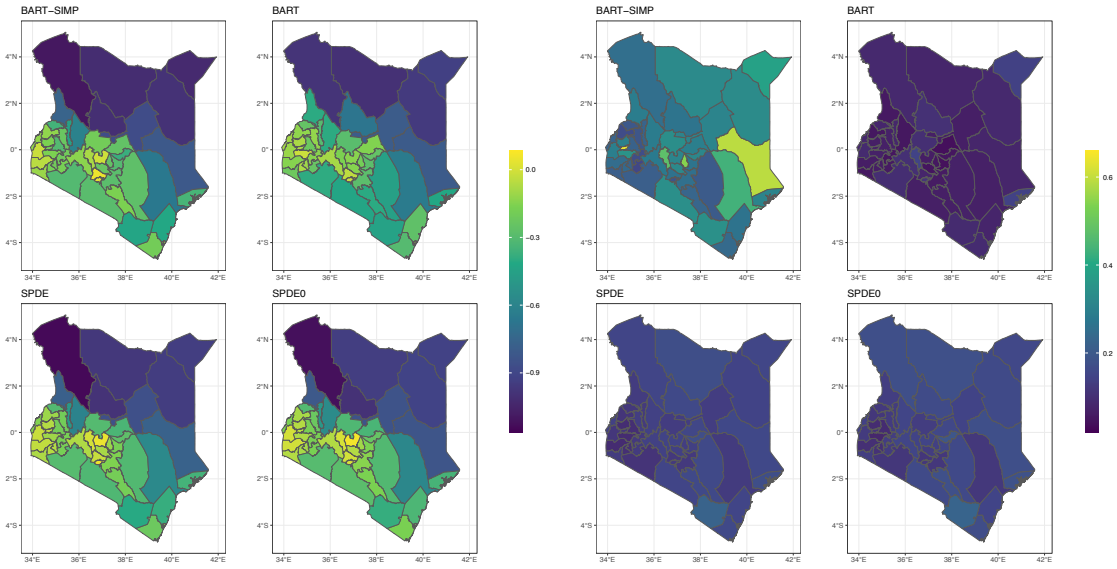

Figure 2.10: Maps of Admin 1 level WHZ posterior median (left) and 90% credible interval lengths (right) for BARTSIMP, BART, SPDE and SPDE0.

2.7.3 Admin 2 Results for the Application Example

Figures 2.11 and 2.12 show maps of posterior median and 95% credible interval lengths for the four methods: BARTSIMP, BART, SPDE and SPDE0. Note that the results are similar to the results for Admin 1 areas, where the point predictions are similar across different methods, while BARTSIMP has larger confidence interval lengths than all other methods. Figures 2.13 to 2.18 plots the posterior median and 95% credible intervals for all 290 Admin 2 enumeration areas. Note that we have split the graphs into six sub-graphs due to the amount of Admin 2 areas in consideration.

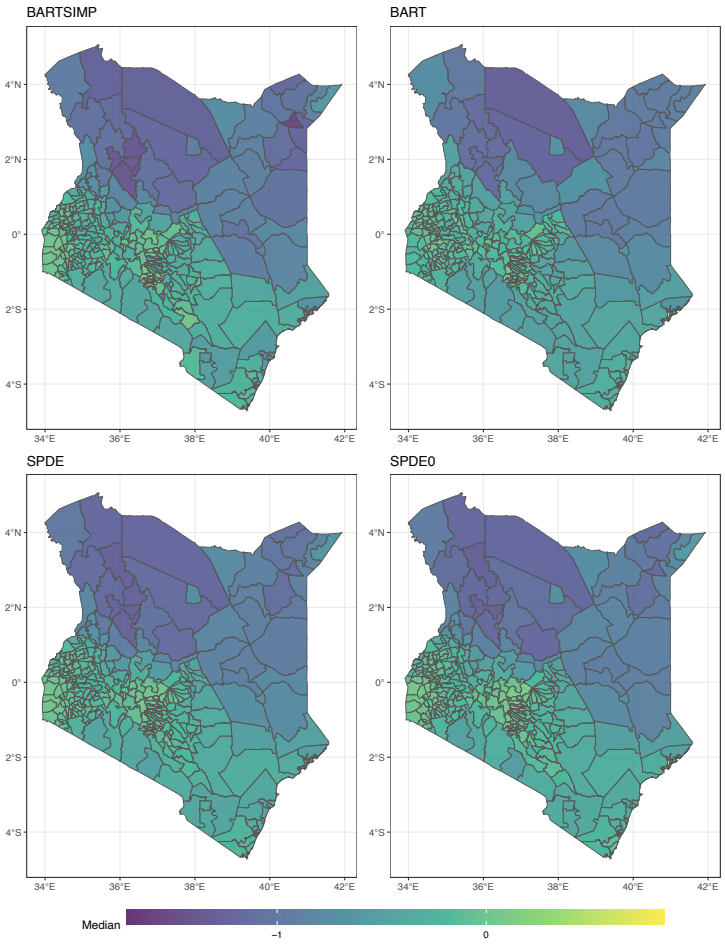

Figure 2.11: Maps of Admin 2 level WHZ posterior median (left) and 95% credible interval lengths (right) for BARTSIMP, BART, SPDE and SPDE0.

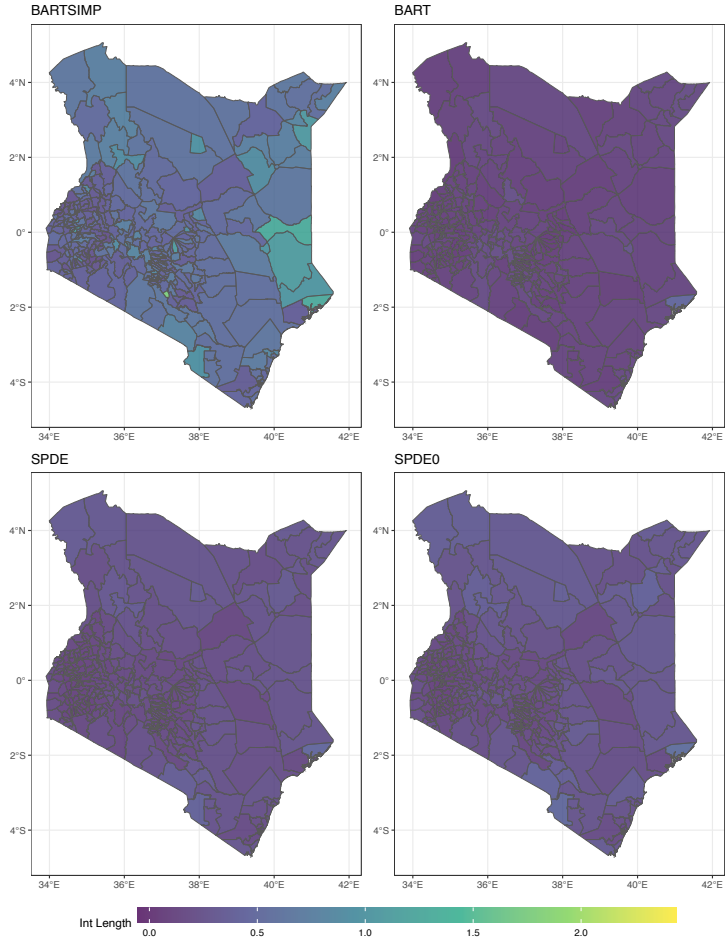

Figure 2.12: Maps of Admin 2 level WHZ posterior median (left) and 95% credible interval lengths (right) for BARTSIMP, BART, SPDE and SPDE0.

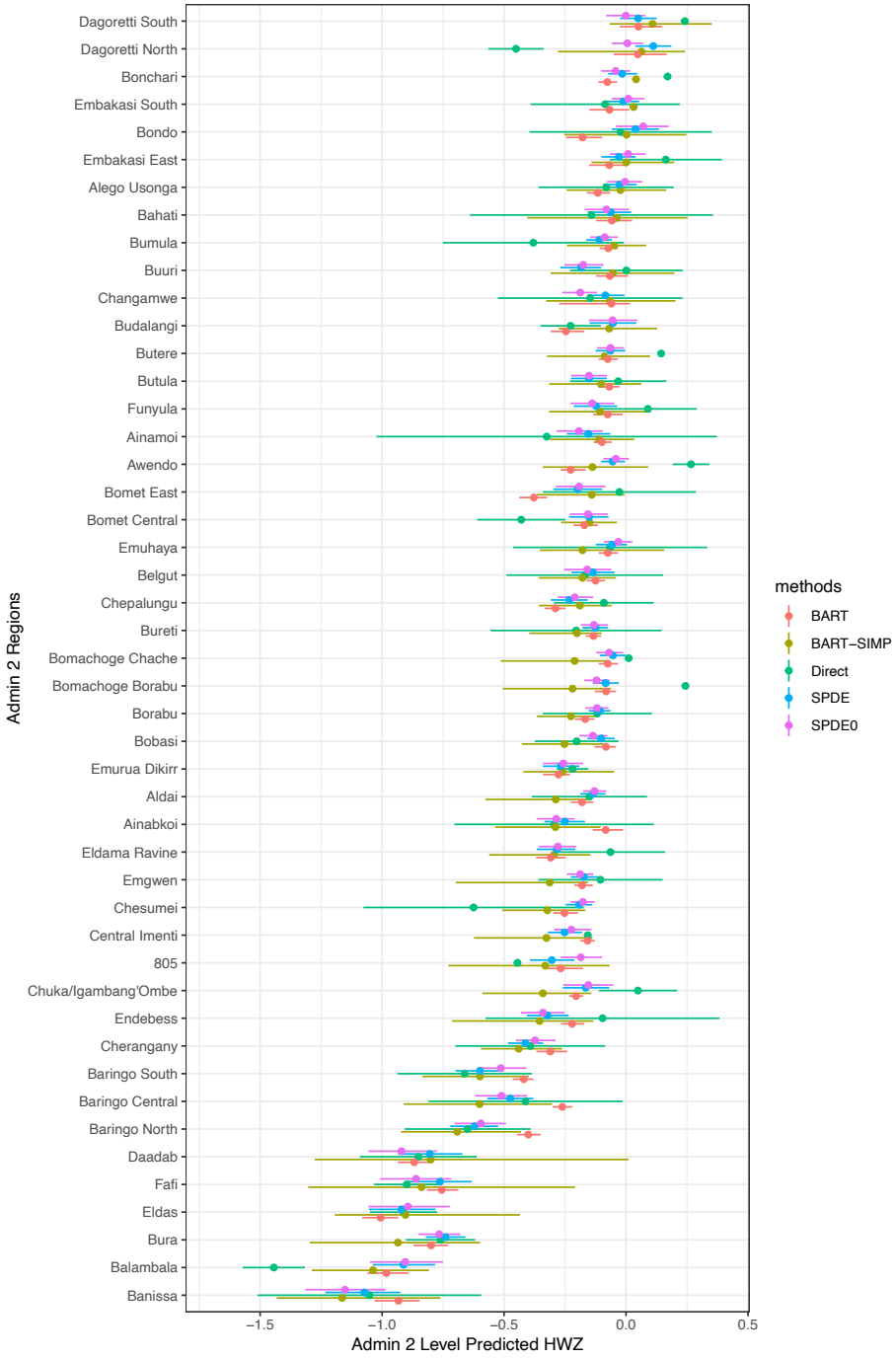

Figure 2.13: The Admin 2 level posterior median and 95% credible/confidence intervals of WHZ for BART, BARTSIMP, SPDE0, SPDE and direct (weighted) estimates. The Admin 1 areas are arranged according to the predicted posterior mean given by BARTSIMP.

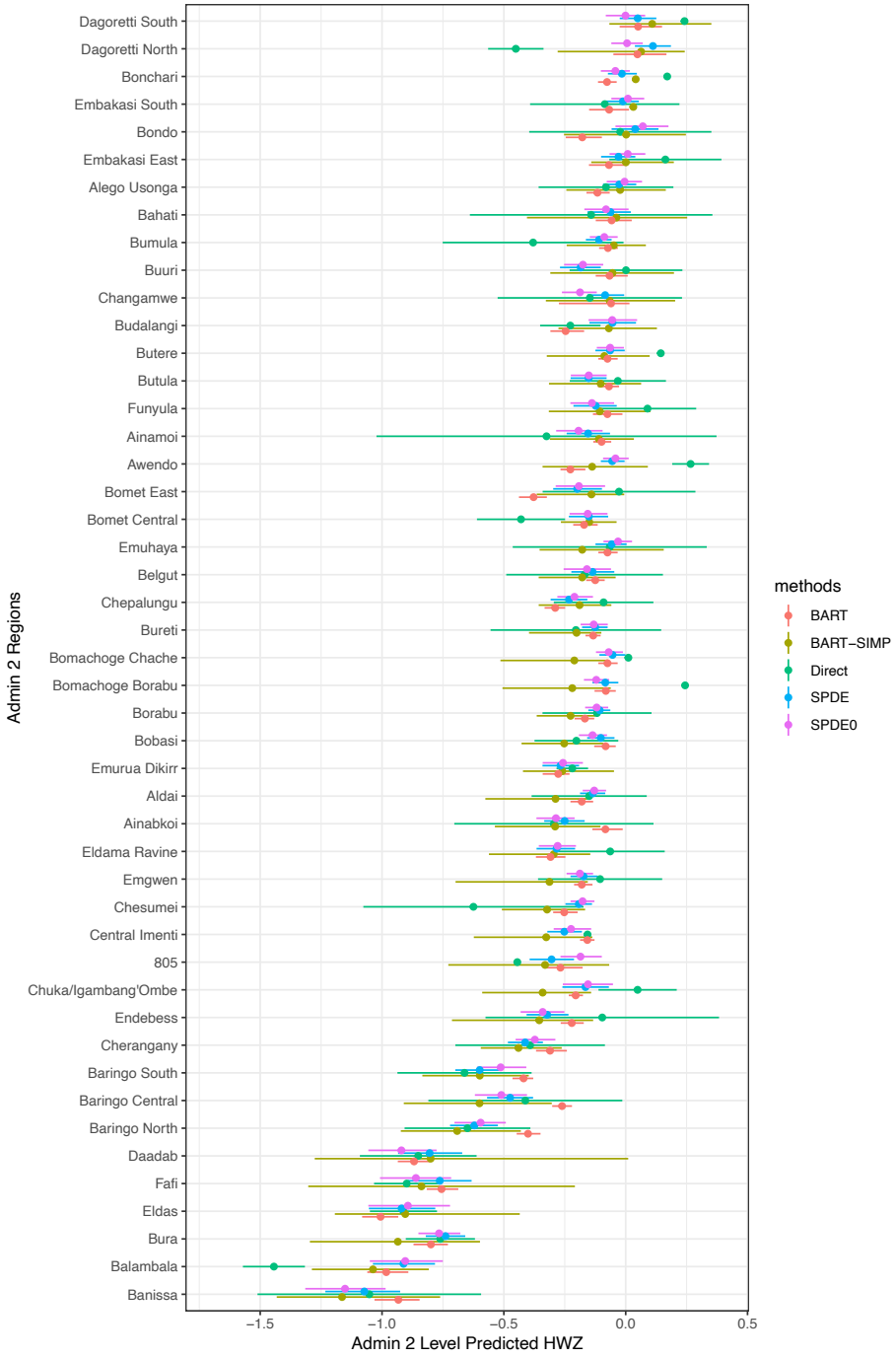

Figure 2.14: The Admin 2 level posterior median and 95% credible/confidence intervals of WHZ for BART, BARTSIMP, SPDE0, SPDE and direct (weighted) estimates. The Admin 1 areas are arranged according to the predicted posterior mean given by BARTSIMP.

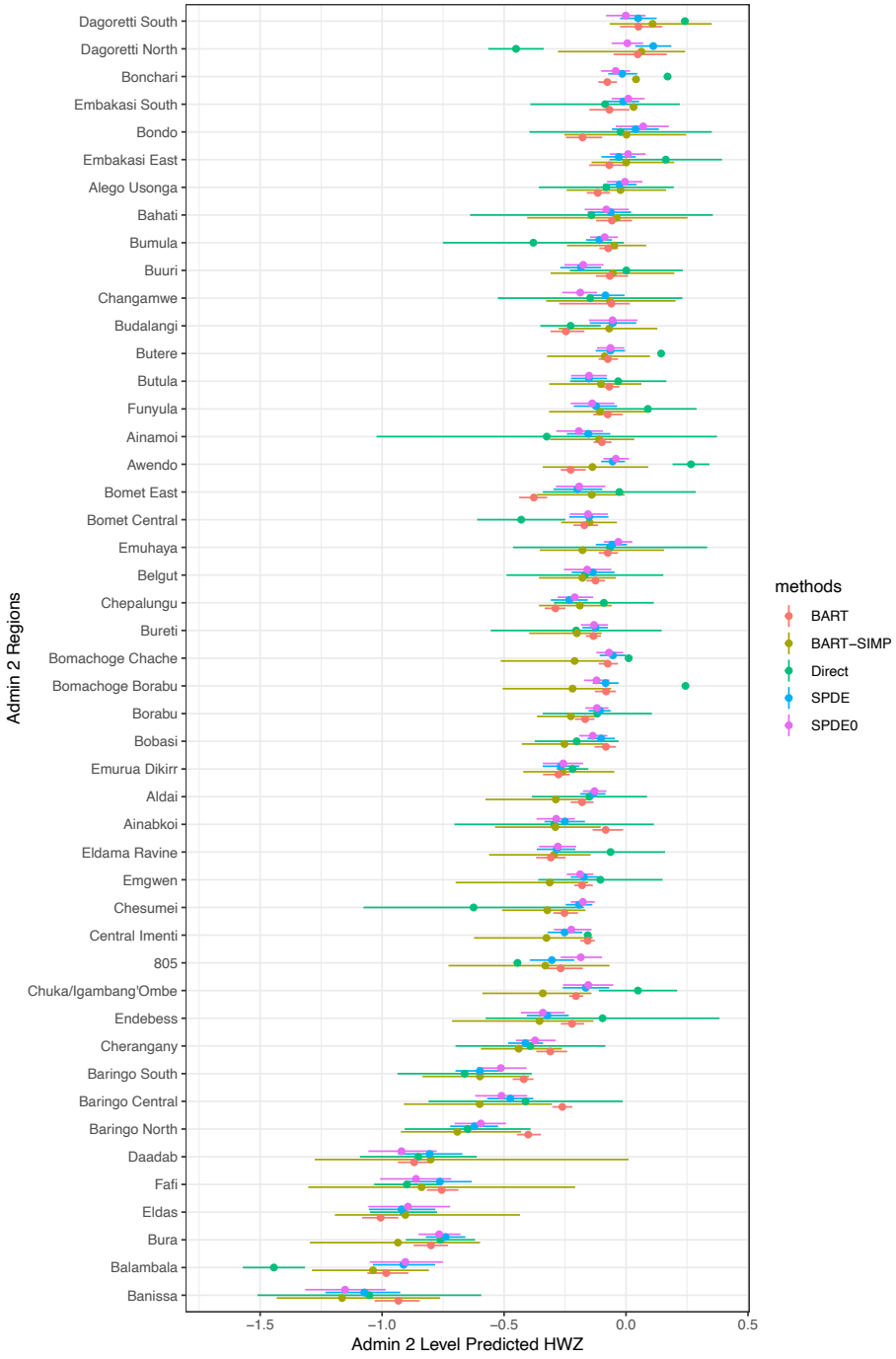

Figure 2.15: The Admin 2 level posterior median and 95% credible/confidence intervals of WHZ for BART, BARTSIMP, SPDE0, SPDE and direct (weighted) estimates. The Admin 1 areas are arranged according to the predicted posterior mean given by BARTSIMP.

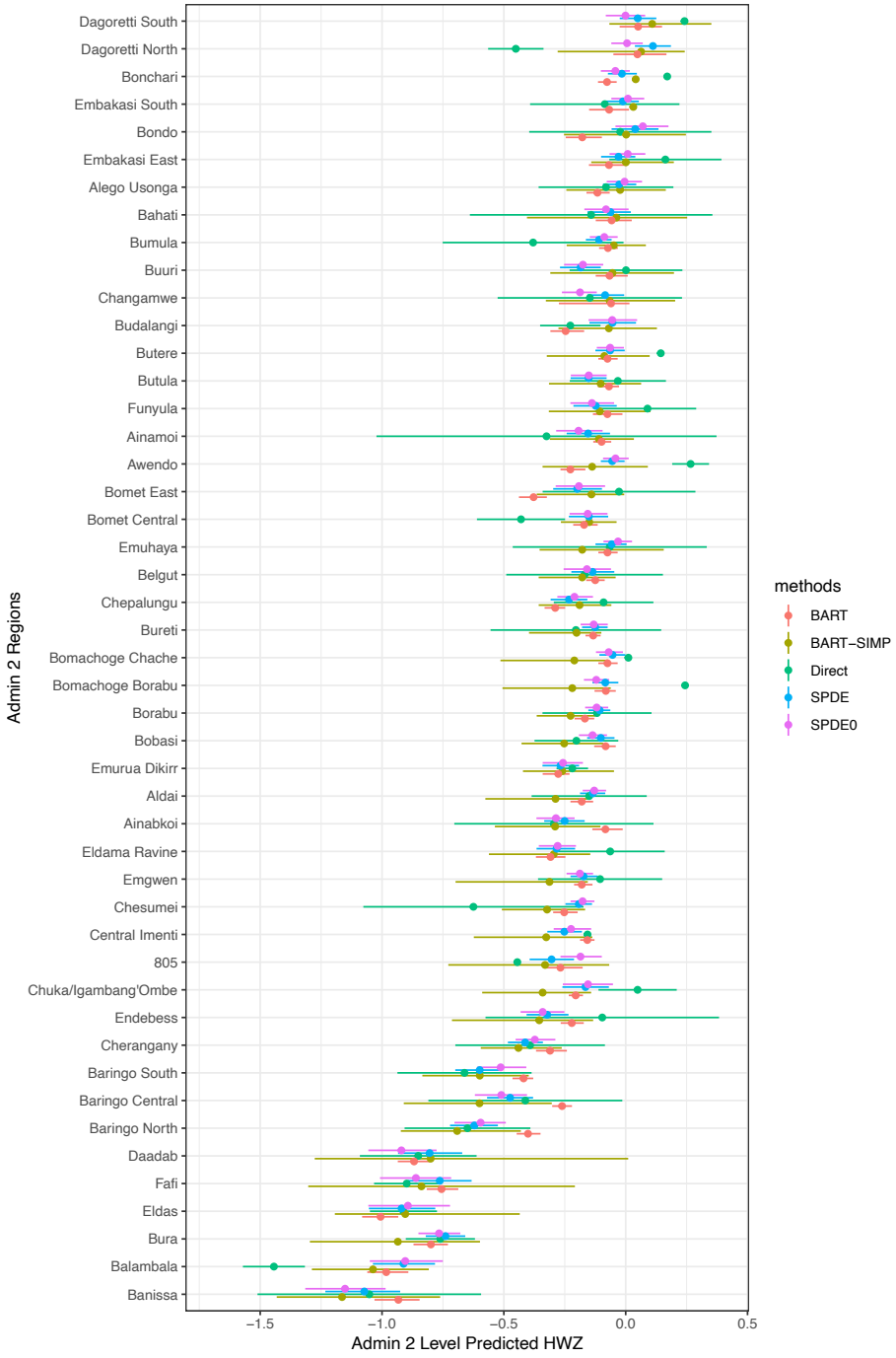

Figure 2.16: The Admin 2 level posterior median and 95% credible/confidence intervals of WHZ for BART, BARTSIMP, SPDE0, SPDE and direct (weighted) estimates. The Admin 1 areas are arranged according to the predicted posterior mean given by BARTSIMP.

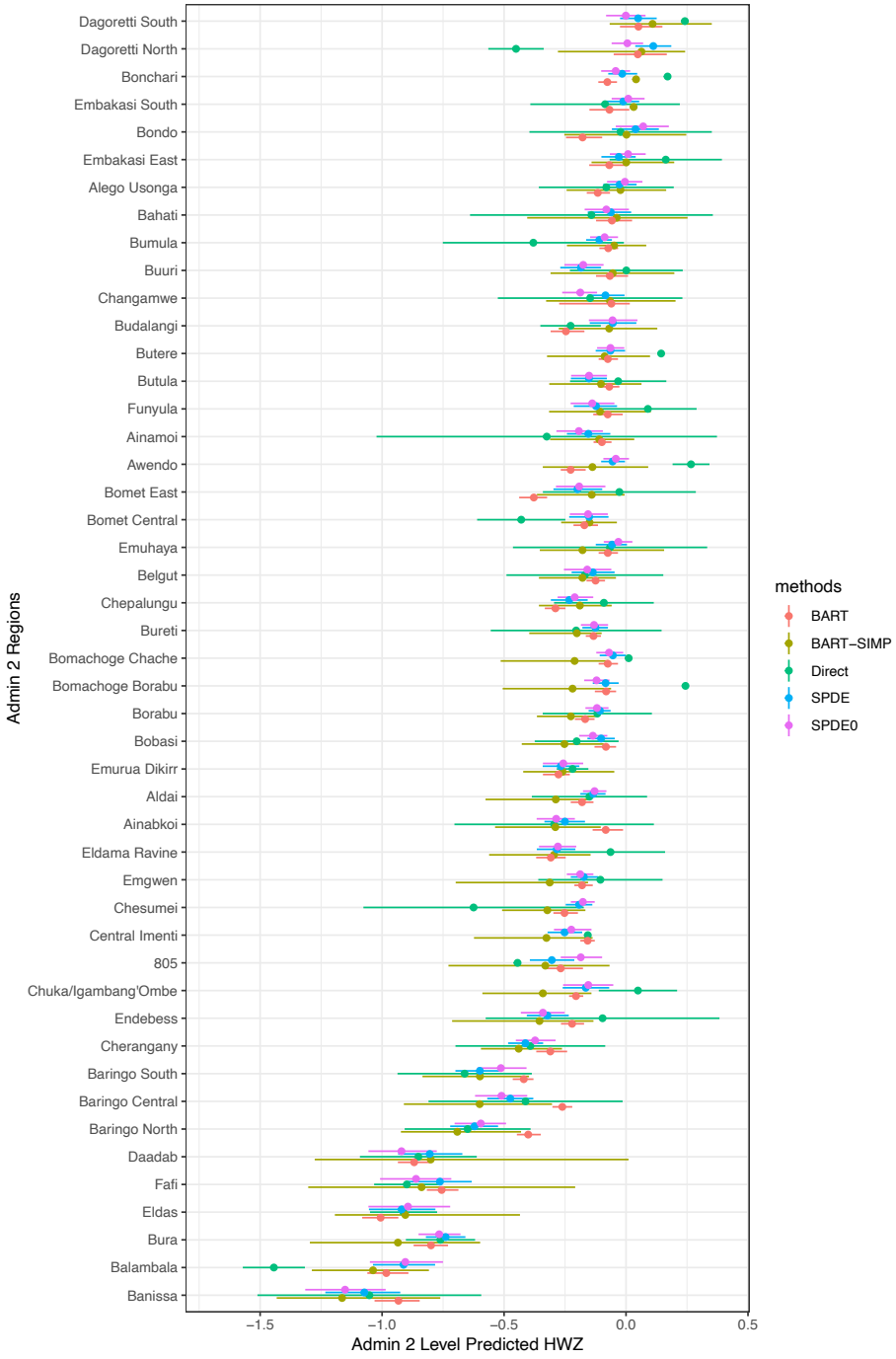

Figure 2.17: The Admin 2 level posterior median and 95% credible/confidence intervals of WHZ for BART, BARTSIMP, SPDE0, SPDE and direct (weighted) estimates. The Admin 1 areas are arranged according to the predicted posterior mean given by BARTSIMP.

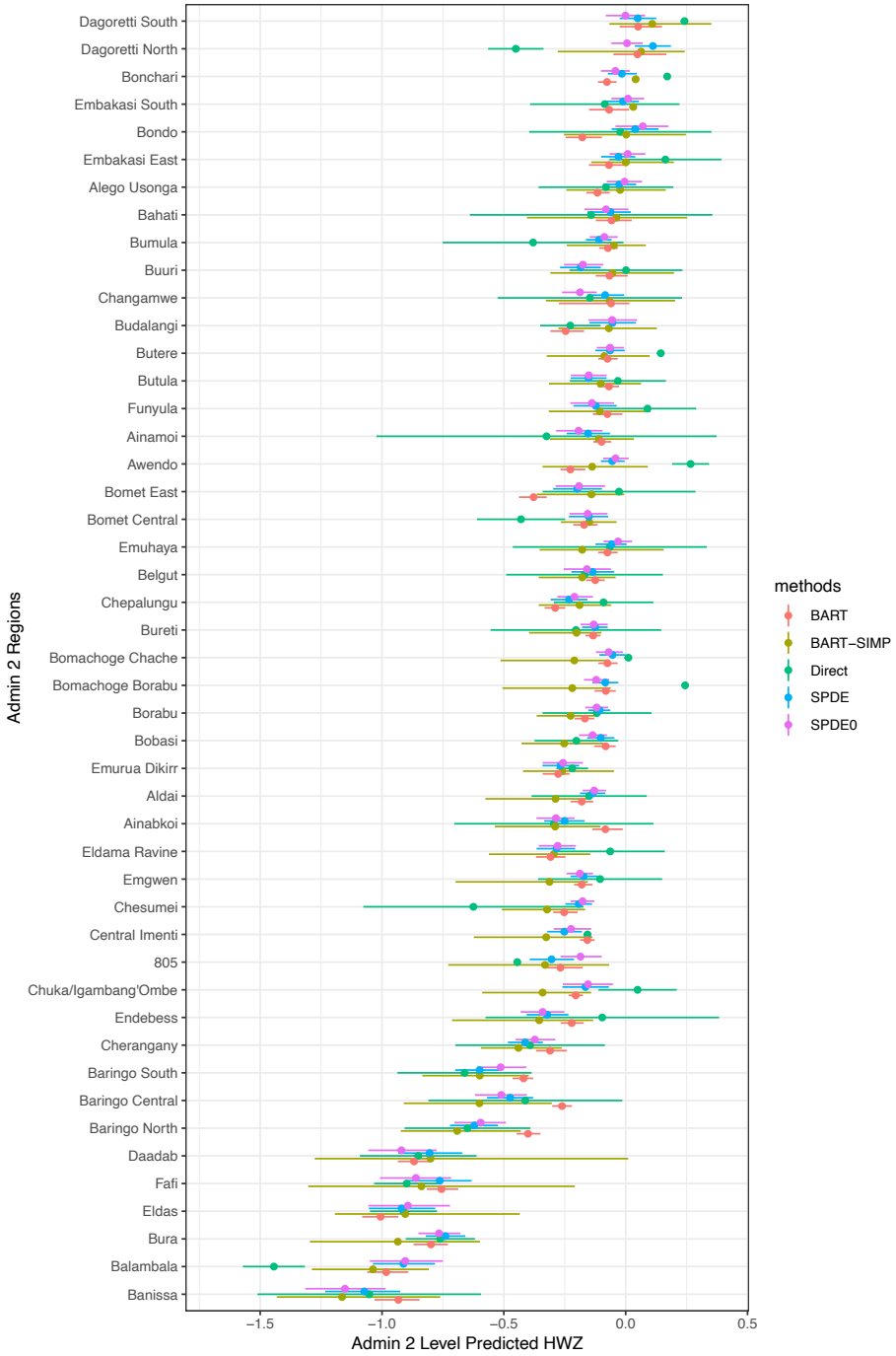

Figure 2.18: The Admin 2 level posterior median and 95% credible/confidence intervals of WHZ for BART, BARTSIMP, SPDE0, SPDE and direct (weighted) estimates. The Admin 1 areas are arranged according to the predicted posterior mean given by BARTSIMP.

2.7.4 Partial dependence

In our example, it is also of interest to study the marginal influence of each variable. Various measures of partial dependence have been proposed in the machine learning literature [21, 60]. For models based on BART, the partial dependence function [53] is a commonly used measure. Let $f(\cdot) : \mathbb{R}^p \to \mathbb{R}$ be a multivariate function defined on p variables $\boldsymbol{x} = (x_1, \ldots, x_p)$. Let $\boldsymbol{x}_t$ denote the index of variables we wish to study, out of the p variables, and let $\boldsymbol{x}_c = \boldsymbol{x}/x_t$ be their compliment. Suppose we have n observations of such multivariate variables. The partial dependence function for $f(\cdot)$ with respect to x_t is defined as

$$f^{pd}(x_t) = \frac{1}{n}\sum_{i=1}^{n} f(x_t, \boldsymbol{x}_{ic}),$$

where $\boldsymbol{x}_{i.}, i = 1, \ldots, n$ represents the i-th observation. For BARTSIMP and BART, we can calculate the partial dependence function of the sum-of-trees function for all six variables in our dataset. Figure 2.20 shows the partial dependence function and its 95% credible interval based on 2000 MCMC samples for the BARTSIMP and BART methods. We observe that for all six variables, BARTSIMP and BART have similar partial dependence patterns that roughly resemble the pattern of the raw data. Substantively, holding all other variables constant, WHZ increases with increasing population density and precipitation, and decreases with increasing average temperature and access to nearest city. Note that BART has narrower credible intervals, especially when the covariate values are more extreme, which support our previous finding that BART tends to underestimate model uncertainty in this setting. We compare the results in Figure 2.20 to the posterior summary of the fixed effects, fitted from an SPDE model to the same dataset. Table 2.9 shows posterior summary of the fixed effects in the SPDE model we fitted for the WHZ dataset. Note that in the table, the population density is positively significant, while covariates including vegetation index and access to nearest city, average temperature are negatively significant. This matches with the results in the partial dependence plot in Figure 2.20.

	Posterior Mean	2.5 Percentile	Median	97.5 Percentile
(Intercept)	-0.61	-0.82	-0.62	-0.30
population density	0.06	0.03	0.06	0.09
vegetation index	-0.01	-0.05	-0.01	0.02
access to nearest city	-0.03	-0.05	-0.03	-0.01
average temperature	-0.12	-0.18	-0.12	-0.06
night time light	0.01	-0.01	0.01	0.03
precipitation	0.02	-0.05	0.02	0.10

Table 2.9: Posterior summary of the fixed effects in the SPDE model.

We note that for the SPDE model, the population density has a positive association, while covariates including access to nearest city, average temperature have negative associations, this is in line with the results shown in Figure 2.20. Additionally, we note that these results are plotted on the original scale of the covariates (e.g., before transformation), which caused the majority of the data points used to generate the line plots to be 'clustered' towards the left end of the x-axis. Due to the log-transformation we applied to a few of the covariates, the partial dependence plots on some of the covariates are gridded unevenly. As an alternative, we also included an alternative version of the partial dependence plots in 2.20 with x on the transformed scale. Please refer to 2.1 in Section 2.2 for detailed information on the scales of the variables, before and after transformation.

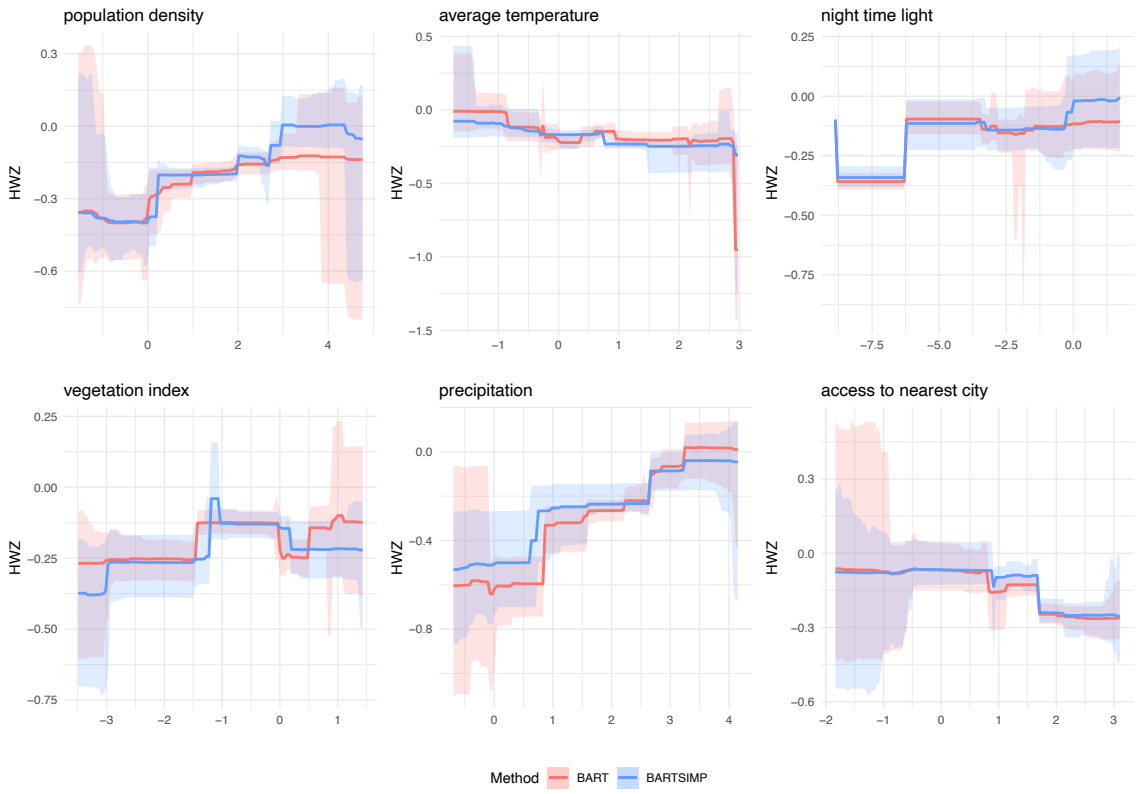

Figure 2.19: Partial dependence function for all six covariates (population density on the transformed scaled, average temperature, night time light, vegetation index, precipitation and access to nearest city) and the 95% pointwise credible interval provided by the BART-SIMP (black) and BART (red) methods.

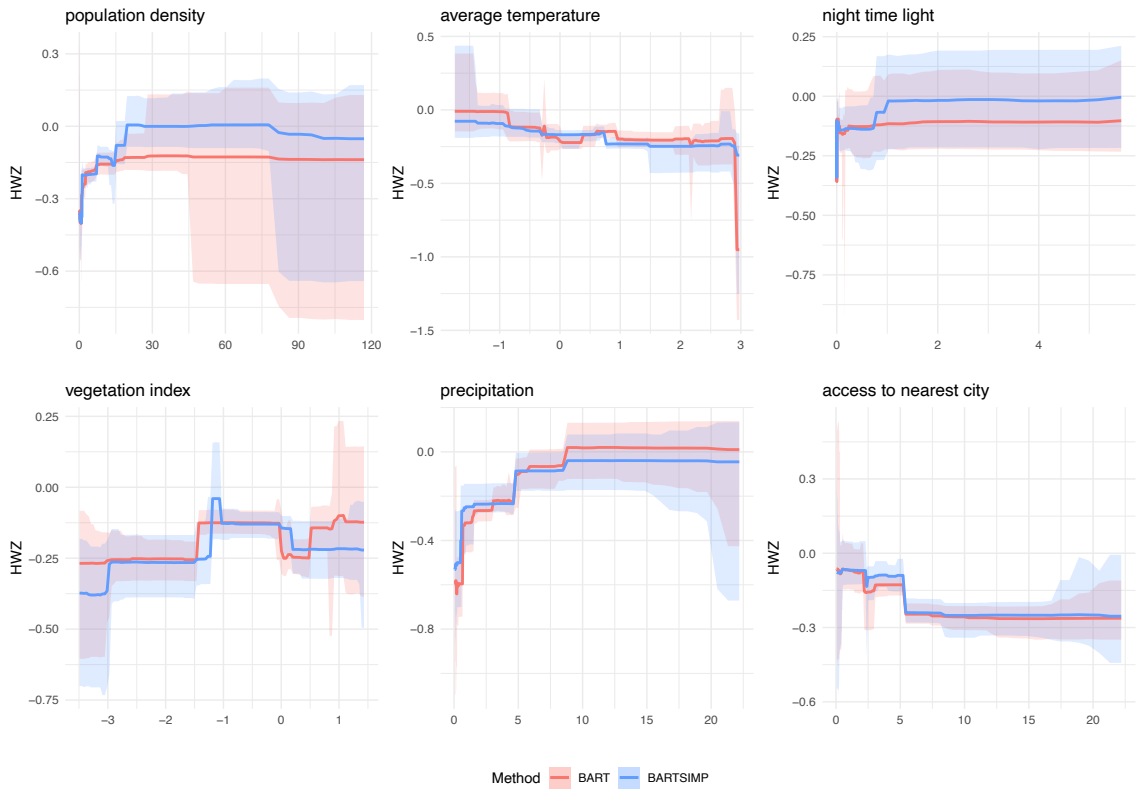

Figure 2.20: Partial dependence function for all six covariates (population density, average temperature, night time light, vegetation index, precipitation and access to nearest city) and the 95% pointwise credible interval provided by the BARTSIMP (black) and BART (red) methods.

2.7.5 Computation Time: Application Examples

datasets	BARTSIMP	BART	SPDE	SPDE0
1	520.08	7.73	0.51	0.48
2	510.90	5.36	0.45	0.53
3	515.22	5.15	0.48	0.51
4	513.30	5.00	0.54	0.52
5	515.84	5.18	0.47	0.50
6	481.30	5.12	0.49	0.48
7	483.38	5.17	0.47	0.46
8	449.94	4.55	0.49	0.52
9	427.91	4.78	0.46	0.58
10	426.24	5.25	0.49	0.40

Table 2.10: Comparison of the computation time in the application example on 10 datasets among the four different methods. The numbers are shown in minutes.

2.8 Discussion

In this chapter, we have proposed BARTSIMP as a novel framework for flexible covariate modeling and prediction for spatial datasets. We incorporated the nonparametric nature of BART into continuous spatial models and leveraged the flexibility of BART to allow nonlinear covariate relationships and interactions. To implement the model we developed a sampling-based inference algorithm for BARTSIMP, based on the INLA-within-MCMC technique.

BARTSIMP has a number of limitations which require more investigation. A cross-validation analysis showed that the BARTSIMP method yields average coverage rates closer to the nominal coverage compared to other methods while having poorer estimation performance than other methods when the covariate signal is weak. This suggests that when the covariate signal in the dataset is not strong compared to the spatial signal then it is

not worth attempting any flexible covariate modeling. This is supported by our simulation studies. We hope our study provides inspiration for future spatial modeling studies with complex covariate patterns. Another potential extension is to consider non-Gaussian likelihood models where the outcome variable is discrete (i.e., counts and proportions) using the Pólyâ Gamma data augmentation technique [127].

The key challenge that arises when one combines spatial models with machine learning techniques, is attaching an appropriate measure of uncertainty. In general, uncertainty estimation is difficult with machine learning techniques, since the bootstrap does not work in many instances. For example, with sparse estimators this occurs because the limiting distribution is complex and may not be continuous, see [39]. When combining machine learning techniques with spatial models, this aspect becomes even more challenging.

Chapter 3

IMPROVEMENTS ON SCALABLE STOCHASTIC BAYESIAN INFERENCE METHODS FOR MULTIVARIATE HAWKES PROCESS

3.1 Introduction

The multivariate Hawkes process (MHP) model [67, 95] is a class of temporal point process models that can capture complex time-event dynamics among multiple objects. Specifically, MHPs demonstrate the *self-* and *mutually-exciting* properties in multidimensional event sequences, where an event occurrence in a certain dimension leads to a higher likelihood of future events appearing in the same or other dimensions. This feature of the models makes MHPs attractive in a wide range of applications, including earth sciences [123], finance [7] and social media analysis [135].

Computational methods for maximum likelihood inference in Hawkes process models include direct maximization of the likelihood function (e.g., see 126) and the expectation-maximization (EM) algorithm (e.g., see 157 and 90), as well as penalized least-squares estimation method of [6] and the variable selection-integrated maximum likelihood method of [18] for sparse interaction settings. In the context of Bayesian inference, some of the algorithms that have been proposed include Markov Chain Monte Carlo algorithms (MCMC) [131, 113, 72, 71, 38], variational approximations [164, 101, 150], sequential Monte Carlo [92], and the maximum *a posteriori* probability estimation using the Expectation-Maximization algorithm (EM) [167]. In addition, theoretical guarantees for nonparametric Bayesian estimation methods of MHPs have been studied in [42] and [149]. One key challenge associated with all these computational approaches is that they do not scale well to large datasets. Specifically, the double summation operation needed to carry out a single likelihood evaluation is typically of time complexity $\mathcal{O}(KN^2)$, where K is the number of dimensions and N is the number of total events. Even in situations where careful implementation can reduce the time complexity to $\mathcal{O}(KN)$ (e.g., for exponential excitation functions), the cost of

this operation can be prohibitive for moderately large datasets. Furthermore, for methods that utilize the branching structure of MHPs, the space complexity is $\mathcal{O}(N^2)$ in all cases. An additional complication is that the calculation of the so-called "compensator" term in the likelihood function might limit our ability to exploit potential conjugacy in the model structure. Standard approximations to the compensator, which are well-justified when maximizing the full likelihood, can have a more serious impact when applied to small datasets, e.g., those arising in the context of algorithms that use subsets of the original data.

Algorithms inspired by stochastic optimization [136] ideas, which approximate the gradient of the objective function through noisy versions evaluated on subsamples, offer an alternative for Bayesian inference on large datasets. Examples of such algorithms include stochastic gradient EM algorithms for finding the posterior mode of a model (e.g., see 27), stochastic gradient variational algorithms (e.g., see 70) and stochastic gradient Hamiltonian Monte Carlo methods (e.g., see 119 and references therein). The use of stochastic gradient methods in the context of MHP models is, nonetheless, limited. Exceptions include [91], who consider the use of stochastic gradient variational inference in the context of a discretized MHP, and [120], who discuss stochastic gradient methods to directly maximize the observed data likelihood.

In this chapter, we discuss the efficient implementation of stochastic gradient EM, stochastic gradient variational approximations, and stochastic gradient Langevin diffusion methods in the context of parametric MHP models, and evaluate various aspects of their performance using both simulated and real datasets. A key contribution is an investigation of a novel approximation technique for the likelihood of the subsamples based on first-order Taylor expansion of the compensator term of the MHP models. We show that this novel approximation can lead to improvements in both point and interval estimation accuracy. For illustration purposes, we focus on intensity functions with exponential excitation functions. However, the insights gained from our experiments can be useful when working with other excitation functions that are proportional to density functions for which a conjugate prior on the unknown parameters is tractable.

Not only is the literature on stochastic gradient methods for Bayesian inference in MHP models limited, but the trade-offs between computational speed and accuracy are not well

understood in this context. For *full-batch* methods (i.e., when using gradients based on the whole dataset rather than subsamples) [169] compares the estimation properties for EM, variational and random-walk MCMC algorithms. This chapter extends this comparative evaluation to algorithms based on stochastic gradient methods. Additionally, we apply our methods to study the market risk dynamics in the Standard & Poor (S&P)'s 500 intraday index prices for its 11 sectors, using all three computation approaches. Our analysis suggests that the 11 sectors can be grouped into four categories. Price movements from sectors within the same category are more likely to be associated with each other, compared to sectors from different categories. Our analysis also shows that the effective range of the interactions between sectors is in the order of at most a few minutes.

3.2 Multivariate Hawkes process models

Let $\mathbf{X} = \{(t_i, d_i) : i = 1, \ldots, n\}$ be a realization from from a marked point process where $t_i \in \mathbb{R}^+$ represents the time at which the i-th event occurs and $d_i \in \{1, \ldots, K\}$ is a mark that represents the dimension in which the event occurs. For example, t_i might represent the time at which user d_i makes a social media post, or the time at which the price of stock d_i drops below a certain threshold. Also, let n be the total number of events in the sequence and let n_k be the number of events in dimension k. Similarly, let $\mathcal{H}_t = \{(t_i, d_i) : t_i < t, t_i \in \mathbf{X}\}$ be the set of events that happened up until time t, and $N^{(k)}(t)$ be the number of events in dimension k that occurred on $[0, t]$. A sequence $\mathbf{X}$ follows a multivariate Hawkes process if the conditional density function on dimension ℓ has the following form:

$$\lambda_\ell(t) \equiv \lim_{h \to 0} \frac{\mathbb{E}[N^{(\ell)}(t+h) - N^{(\ell)}(t) \mid \mathcal{H}_t]}{h} = \mu_\ell + \sum_{k=1}^{K} \sum_{t_i < t, d_i = k} \phi_{k,\ell}\left(t - t_i\right), \tag{3.1}$$

where $\mu_\ell > 0$ is the background intensity for dimension ℓ, and $\phi_{k,\ell}(\cdot) : \mathbf{R}^+ \to \mathbf{R}^+$ is the excitation function that controls how previous events in dimension ℓ affect the occurrence of new events in dimension k.

Common modeling choices for the excitation function include the exponential decay function where $\phi_{k,\ell}(\Delta) = \alpha_{k,\ell}\beta_{k,\ell}e^{-\beta_{k,\ell}\Delta}$ for $\Delta \geq 0$, and the power law decay function where where $\phi_{k,\ell}(\Delta) = \frac{\alpha_{k,\ell}\beta_{k,\ell}}{(1+\beta_{k,\ell}\Delta)^{1+\gamma_{k,\ell}}}$ for $\Delta \geq 0$. For illustration purposes, in this chapter,

we focus on the exponential decay function. In that case, the parameter $\alpha_{k,\ell}$ controls the importance of events from dimension k on the appearance of events in dimension ℓ, and $\beta_{k,\ell}$ controls the magnitude of exponential decay of the instant change associated with a new event.

Let $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu})$ denote the vector of all model parameters. Using standard theory for point processes (e.g., see [33]), the observed log-likelihood associated with a Hawkes process can be written as

$$\begin{aligned}\mathcal{L}(\mathbf{X} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}) &= \sum_{\ell=1}^{K} \sum_{d_i=\ell} \log \lambda_\ell(t_i) - \sum_{\ell=1}^{K} \int_0^T \lambda_\ell(s) ds \\ &= \sum_{\ell=1}^{K} \sum_{d_i=\ell} \log \left(\mu_\ell + \sum_{k=1}^{K} \sum_{\substack{j<i \\ d_j=k, d_i=\ell}} \alpha_{k,\ell} \beta_{k,\ell} e^{-\beta_{k,\ell}(t_i - t_j)} \right) - \sum_{\ell=1}^{K} \mu_\ell T \\ &\qquad - \sum_{k=1}^{K} \sum_{\ell=1}^{K} \alpha_{k,\ell} \left[n_k - \sum_{d_i=k} \exp\left(-\beta_{k,\ell}(T - t_i)\right) \right].\end{aligned} \tag{3.2}$$

The MHP can also be obtained as a multidimensional Poisson cluster process in which each point is considered an "inmigrant" or an "offspring" [68, 103, 170, 131]. We use the lower-triangular $n \times n$ binary matrix $\mathbf{B}$ to represent the latent branching structure of the events, where each row contains one and only one non-zero entry. For the strictly lower-triangular entries on the matrix, $B_{ij} = 1$ indicates that the i-th event can be viewed as an offspring of the j-th event. On the other hand, for the diagonal entries of the matrix, $B_{ii} = 1$ indicates that the i-th event is an immigrant. Each immigrant independently generates a cluster of offsprings that can further generate newer generations of offspring.

The branching structure, which is typically latent and unobservable, allows us to decouple the complex observed likelihood into factorizable terms and design simpler computational algorithms. The complete data log-likelihood, defined as the joint log-likelihood of

the observed data and the branching structure $\mathbf{B}$, has the following form :

$$\mathcal{L}(\mathbf{X}, \mathbf{B} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}) = \sum_{\ell=1}^{K} |I_\ell| \log \mu_\ell + \sum_{k=1}^{K}\sum_{\ell=1}^{K} \left[|O_{k,\ell}| \left(\log \alpha_{k,\ell} + \log \beta_{k,\ell}\right) - \beta_{k,\ell} \sum_{\substack{j<i \\ d_j=k, d_i=l}} B_{ij} \left(t_i - t_j\right) \right] - \sum_{\ell=1}^{K} \mu_\ell T - \sum_{k=1}^{K}\sum_{\ell=1}^{K} \alpha_{k,\ell} \left[n_k - \sum_{d_i=k} \exp\left(-\beta_{k,\ell}\left(T - t_i\right)\right) \right], \tag{3.3}$$

where $|I_\ell| = \sum_{\substack{1 \le i \le n \\ d_i = \ell}} B_{ii}$ is the number of immigrants for dimension ℓ, and $|O_{k,\ell}| = \sum_{\substack{j<i \\ d_j=k, d_i=\ell}} B_{ij}$ is the number of descendants on dimension k that are an offspring of an event on dimension j.

3.2.1 Approximation for the data likelihood

The expressions for the observed and complete data likelihood in (3.2) and (3.3) share the same term

$$\Upsilon = \sum_{\ell=1}^{K} \int_0^T \lambda_\ell(s) ds = \sum_{\ell=1}^{K} \mu_\ell T + \sum_{k=1}^{K}\sum_{\ell=1}^{K} \alpha_{k,\ell} \left[\sum_{d_i=k} \left(1 - \exp\left\{-\beta_{k,\ell}\left(T - t_i\right)\right\}\right) \right]. \tag{3.4}$$

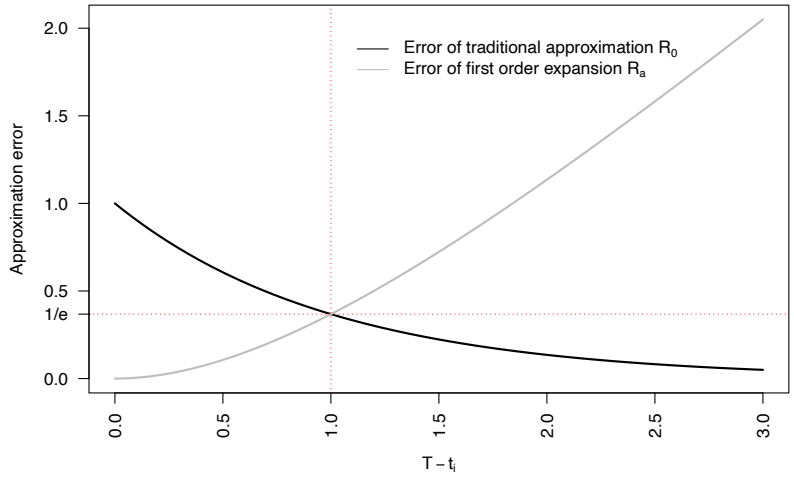

Figure 3.1: Trade off between approximation errors for the compensator for $\beta_{k,\ell} = 1$.

The integral $\int_0^T \lambda_\ell(s)ds$ is known as the compensator for the conditional density function $\lambda_\ell(t)$. The compensator guarantees that there are infinitely many 'none events' between observations on the finite temporal region $[0, T]$ [105]. The form of the compensator causes a number of computational challenges for designing scalable inference algorithms for MHP models (for a discussion see 90, 143, 72). A common approach used to avoid these challenges is to use the approximation technique introduced in [90]:

$$\alpha_{k,\ell}\left[1 - \exp\left(-\beta_{k,\ell}\left(T - t_i\right)\right)\right] \approx \alpha_{k,\ell}, \tag{3.5}$$

for all $k, \ell = 1, \ldots, K$. Clearly, this approximation is quite accurate for large values $T - t_i$. Therefore, if the process is stationary and T is "large" (ensuring that most observation are far away from T), the error $R_0(T - t_i) = \exp\left\{-\beta_{k,\ell}(T - t_i)\right\}$ is negligible for most i. However, for observations that are "close" to the boundary T, the approximation is quite poor. This will be an important consideration when designing algorithms for the MHP model that rely on subsamples because, in that setting, the equivalent quantity to T will be small and the edge effects cannot necessarily be assumed to be negligible.

To address this issue, we propose to use a different approximation for observations that are close to T. This alternative approximation is based on a first order Taylor expansion of the exponential function, so that

$$\alpha_{k,\ell}\left[1 - \exp\left\{-\beta_{k,\ell}\left(T - t_i\right)\right\}\right] \approx \alpha_{k,\ell}\left(1 - \left[1 - \beta_{k,\ell}(T - t_i)\right]\right) = \alpha_{k,\ell}\beta_{k,\ell}(T - t_i) \tag{3.6}$$

The error associated with this approximation is $R_a(T - t_i) = \sum_{n=2}^{\infty} \frac{(-\beta_{k,\ell}(T-t_i))^n}{n!} = \exp\left\{-\beta_{k,\ell}\left(T - t_i\right)\right\} - 1 + \beta_{k,\ell}(T - t_i)$. Hence, as opposed to (3.5), the approximation in (3.6) is accurate for small $T - t_i$. Furthermore, R_a is smaller than R_0 if and only if $T - t_i > 1/\beta_{k,\ell}$ (see Figure 3.1). This suggests dividing $\mathbf{X}$ into two parts, based on whether the data points are observed within a predetermined threshold $(T - \delta, T]$ where $\delta = 1/\beta_{k,\ell}$, so that:

$$\alpha_{k,\ell}\left[\sum_{d_i=k}\left(1 - \exp\left\{-\beta_{k,\ell}\left(T - t_i\right)\right\}\right)\right] \approx \alpha_{k,\ell}\left[n_k - \sum_{0\le T-t_i<\delta, d_i=k}\left[1 - \beta_{k,\ell}(T - t_i)\right]\right], \tag{3.7}$$

for all $k, \ell = 1, \ldots, K$. We call this the boundary-corrected approximation. One of its key advantages that it still allows us to exploit conjugacies in the definition of the model while

providing a more accurate approximation for observations close to T. Please see Section 3.3 for additional details.

Note that, under our proposed approximation, the worst-case absolute error in the likelihood is bounded above by $n_{k,\ell}/e$. Furthermore, the approximation error is guaranteed to always be smaller than that from the original approximation in [90]. To quantify the error caused by approximation, we propose the following definition for the average loss of information:

Definition 3.2.1. ***(Information loss ratio.)*** *Given the time domain $(0, T]$ and approximation threshold δ, We define $\varphi_{k,\ell}(\delta, T)$ as the information loss ratio for approximating $\Lambda_{k,\ell}(\cdot)$, which has the following formula:*

$$\varphi_{k,\ell}(\delta, T) = \frac{1}{T} \int_0^T \left[1 - \frac{1 - \exp\left(-\beta_{k,\ell}(T-t)\right)}{\mathbf{1}(0 < t \leq T - \delta) + [\beta_{k,\ell}(T-t)] \cdot \mathbf{1}(T - \delta < t \leq T)} \right] dt. \tag{3.8}$$

It can be shown that the average loss of information, which is defined in, due to the approximation is negligible as $T \to \infty$. Furthermore, we can interpret $\varphi_{k,\ell}(\delta, T)$ as follows: suppose t is a time event on dimension k, $\varphi_{k,\ell}(\delta, T)$ is one minus the ratio of t's contribution to the compensator (sans the common term $\mu_k T$), integrated over a uniform distribution for t on $[0, T]$. The use of a uniform distribution here seems appropriate under the assumption that the MHP is stationary.

Theorem 3.2.1. *Assuming $\max_{k,\ell} |\beta_{k,\ell}| \leq M$ and δ is fixed, the information loss ratio converges to zero as $T \to \infty$, i.e.*

$$\lim_{T \to \infty} \max_{1 \leq k, \ell \leq K} |\varphi_{k,\ell}(\delta, T)| = 0.$$

Proof. We proved the theorem by showing that $\lim_{T\to\infty} |\varphi_{k,\ell}(\delta, T)| = 0$ for all $k, \ell = 1, \ldots, K$. To derive an upper bound on $|\varphi_{k,\ell}(\delta, T)|$, we start by breaking down the integration domain into two parts: $(0, T - \delta]$ and $(T - \delta, T]$. For the first part, we have:

$$\begin{aligned} \left| \frac{1}{T} \int_0^{T-\delta} \left[1 - (1 - \exp(-\beta_{k,\ell}(T-t)))\right] dt \right| &= \left| \frac{1}{T} \int_0^{T-\delta} \exp(-\beta_{k,\ell} t) dt \right| \\ &= \frac{1}{\beta_{k,\ell} T} \left[1 - \exp(-\beta_{k,\ell}(T-\delta))\right]. \end{aligned}$$

For the second part, we have:

$$\begin{aligned}\left|\frac{1}{T}\int_{T-\delta}^{T}\left[1-\frac{1-\exp\left(-\beta_{k,\ell}(T-t)\right)}{[\beta_{k,\ell}(T-t)]}\right]dt\right| &= \frac{1}{\beta_{k,\ell}T}\int_{0}^{\beta_{k,\ell}\delta}\frac{s-1-\exp(-s)}{s}ds \\ &= \frac{1}{\beta_{k,\ell}T}\int_{0}^{\beta_{k,\ell}\delta}\frac{\exp(-2\Delta s)s^2}{2s}ds, \quad (0\leq\Delta s\leq s) \\ &\leq \frac{1}{\beta_{k,\ell}T}\int_{0}^{\beta_{k,\ell}\delta}\frac{s}{2}ds = \frac{\beta_{k,\ell}\delta^2}{4T}.\end{aligned}$$

Assuming δ is fixed and $\beta_{k,\ell}$ is bounded by a fixed constant over $k,\ell = 1,\ldots,K$, both parts converge to zero as $T\to\infty$. As $|\varphi_{k,\ell}(\delta,T)|$ is upper bounded by the two parts above, we showed that $\lim_{T\to\infty}|\varphi_{k,\ell}(\delta,T)| = 0$. □

3.2.2 Prior distributions

Bayesian inference for the MHP requires that we specify priors for the unknown parameters $(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\mu})$. For the baseline intensities we set

$$\mu_\ell \mid a_\ell, b_\ell \overset{i.i.d}{\sim} \text{Gamma}\left(a_\ell, b_\ell\right),$$

which is conditionally conjugate given the branching structure $\mathbf{B}$. Similarly, under the exponential decay functions we use

$$\begin{aligned}\alpha_{k,\ell} \mid e_{k,\ell}, f_{k,\ell} &\overset{i.i.d}{\sim} \text{Gamma}\left(e_{k,\ell}, f_{k,\ell}\right), \\ \beta_{k,\ell} \mid w_{k,\ell}, s_{k,\ell} &\overset{i.i.d}{\sim} \text{Gamma}\left(w_{k,\ell}, s_{k,\ell}\right)\end{aligned}$$

which are also conditionally conjugate.

3.3 Computational methods

3.3.1 Preliminaries

In this section, we describe three stochastic gradient algorithms for MHP models based on the EM, variational inference and an Markov chain Monte Carlo algorithm based on Langevin dynamics algorithm, respectively. Before delving into the details of each algorithm, we discuss three issues that are relevant to the design of all of them.

The first issue refers to how to define the subsamples used to compute the gradient at each iteration. A common approach for regression models is to randomly select independent

observations. However, the temporal dependence in event sequences makes this approach inappropriate for MHP models. Instead, our subsamples consist of all observations contained in the random interval $[T_0, T_0 + \kappa T]$, where we uniformly sample T_0 on $[0, (1-\kappa)T]$ and $\kappa \in (0, 1]$ corresponds to the relative size of the subsample. Similar strategies have been applied to developing stochastic gradient variational algorithms for hidden Markov models [51] and stochastic block models [62].

The second issue relates to the selection of the learning rate ρ_r for the algorithms, which controls how fast the information from the stochastic gradient accumulates. It is well known (e.g., see 136) that the following conditions lead to convergence towards a local optima:

$$\sum_{r=1}^{\infty} \rho_r = \infty, \qquad \sum_{r=1}^{\infty} \rho_r^2 < \infty. \tag{3.9}$$

In the following analysis, we apply the commonly used update schedule for ρ_r, outlined in [162]:

$$\rho_r = \rho_0 (r + \tau_1)^{-\tau_2}, \tag{3.10}$$

where ρ_0 is a common scaling factor, $\tau_2 \in (0.5, 1]$ is the forgetting rate that controls the exponential decay rate, and $\tau_1 \geq 0$ is the delay parameter that downweights early iterations. In our numerical experiments, we investigate the impact of specific choices of τ_1 and τ_2 on the results.

The third issue relates to the use of approximation techniques. We propose to use (3.7) to approximate the likelihood only for the stochastic gradient EM and variational inference algorithms since (conditional) conjugacy is important for developing these algorithms. In contrast, for stochastic gradient Langevin dynamics, we propose to use the exact likelihood in (3.2). For simplicity, we will refer to the algorithms with approximation as their 'boundary-corrected' versions, and we only show the update formula based on the 'common approximation approach' in this section. Additionally, we want to point out that the exponential decay function that we are using allows us to update $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$ using the exact likelihood formula, and we will only consider the approximation when we update $\boldsymbol{\beta}$.

3.3.2 Stochastic Gradient EM algorithm for posterior mode finding

The expectation-maximization (EM) algorithm [37] is an iterative maximization algorithm that is commonly used for latent variable models, especially in cases where knowledge of the latent variables simplifies the likelihood function. For Bayesian models, it can be used for maximum *a posteriori* probability estimation for the model parameters. Let $\mathbf{X}$ be the observed dataset of size N, $\boldsymbol{\theta}$ be the set of model parameters to be estimated, and $\mathbf{B}$ be the set of latent branching structure variables, and denote $(\mathbf{X}, \mathbf{B})$ as the complete dataset. We further assume that the distribution of the complete dataset belongs to the following exponential family:

$$l(\mathbf{X}, \mathbf{B} \mid \boldsymbol{\theta}) = A(\mathbf{X}, \mathbf{B}) \exp\left(\boldsymbol{\phi}(\boldsymbol{\theta})^\intercal \boldsymbol{s}(\mathbf{X}, \mathbf{B}) - \psi(\boldsymbol{\theta})\right), \tag{3.11}$$

where $\boldsymbol{s}(\mathbf{X}, \mathbf{B})$ is the vector of sufficient statistics for the complete data model and $\boldsymbol{\phi}(\boldsymbol{\theta})$ is the canonical form of the vector of parameters and $\boldsymbol{\phi}(\cdot)$ represents a one-to-one transformation.

In the context of Bayesian models, the EM algorithm can be used to obtain the maximum a posteriori (MAP) estimate (e.g., see [98]). Starting from an initial guess of the model parameters $\boldsymbol{\theta}^{(0)}$, the EM algorithm alternatively carries out the following two steps until convergence:

- In the 'E-step', the algorithm estimates the "marginal" sufficient statistics based on the expected value $\hat{\boldsymbol{s}}^{(r)} := \mathrm{E}_{\mathbf{B}|\mathbf{X},\boldsymbol{\theta}^{(r)}}\left[\boldsymbol{s}(\mathbf{X}, \mathbf{B})\right]$.

- In the 'M-step', the algorithm updates the model parameter as the maximizer of the Q function:

  $$\boldsymbol{\theta}^{(r+1)} = \arg\min_{\boldsymbol{\theta}} \left[\boldsymbol{\phi}(\boldsymbol{\theta})^\intercal \hat{\boldsymbol{s}}^{(r)} + \log p(\boldsymbol{\theta})\right].$$

  where $p(\boldsymbol{\theta})$ denotes the prior on $\boldsymbol{\theta}$.

Note that the expectation calculation in the E-step update requires a pass through the whole dataset. As we discussed in the introduction, this can be challenging in very large datasets. The stochastic gradient EM (SGEM) algorithm [25] addresses this challenge by approximating the marginal sufficient statistics with an estimate based on randomly sampled

mini-batches. We let $\mathbf{X}^{(r)}$ denote a subsample of size n (and respectively, we let $\mathbf{B}^{(r)}$ be the set of branching structure that corresponds to the selected subsample). For the stochastic E-step, the SGEM updates the estimated sufficient statistics $\hat{\boldsymbol{s}}^{(r+1)}$ as a linear combination of the previous update and a new estimate of the sufficient statistics based on the random subsample and the current model parameter:

$$\hat{\boldsymbol{s}}^{(r+1)} = (1-\rho_r)\hat{\boldsymbol{s}}^{(r)} + \rho_r \kappa^{-1} \mathrm{E}_{\mathbf{B}^{(r+1)}|\mathbf{X}^{(r+1)},\boldsymbol{\theta}^{(r)}}[\boldsymbol{s}(\mathbf{X}^{(r+1)}, \mathbf{B}^{(r+1)})].$$

where ρ_r is given in (3.10). Because of the way we select the subsamples, $\kappa^{-1}\mathrm{E}_{\mathbf{B}^{(r+1)}|\boldsymbol{X}^{(r+1)},\boldsymbol{\theta}^{(r)}}[\boldsymbol{s}(\mathbf{X}^{(r+1)}, \mathbf{B}^{(r-$ is an unbiased estimate of the sufficient statistics of the model based on the whole dataset. In the following M-step, the SGEM algorithm maximizes the Q function

$$\boldsymbol{\theta}^{(r+1)} = \arg\max_{\boldsymbol{\theta}} \left[\boldsymbol{\phi}(\boldsymbol{\theta})^\intercal \hat{\boldsymbol{s}}^{(r+1)} + \log p(\boldsymbol{\theta}) \right].$$

In the case of the MHP model with exponential excitation functions, $\boldsymbol{\theta}^{(r)} = (\boldsymbol{\mu}^{(r)}, \boldsymbol{\alpha}^{(r)}, \boldsymbol{\beta}^{(r)})$ and the expectation in the E-step is computed with respect to the probabilities

$$p_{i,j}^{(r)} := p\left(\mathbf{B}_{i,j}^{(r)} = 1, \mathbf{B}_{i,-j}^{(r)} = 0 \mid \boldsymbol{\mu}^{(r)}, \boldsymbol{\alpha}^{(r)}, \boldsymbol{\beta}^{(r)}, \mathbf{X}^{(r)}\right) \propto \begin{cases} \mu_{d_i}^{(r)} & \text{if } j = i, \\ \alpha_{d_j,d_i}^{(r)} \beta_{d_j,d_i}^{(r)} \exp(-\beta_{d_j,d_i}^{(r)}(t_i - t_j)) & \text{if } j < i, \\ 0 & \text{if } j > i. \end{cases} \tag{3.12}$$

for $i = 2, \ldots, n$, the negative subindex stands for all other possible except the one, and $p_{1,1}^{(r)} := 1$. Then, the vector of expected sufficient statistics of the complete data likelihood evaluated at iteration r

$$\left(s_{\mu,\ell,1}^{(r)}, s_{\mu,\ell,2}^{(r)}, s_{\alpha,k,\ell,1}^{(r)}, s_{\alpha,k,\ell,2}^{(r)}, s_{\beta,k,\ell,1}^{(r)}, s_{\beta,k,\ell,2}^{(r)}\right),$$

is updated as

$$
\begin{aligned}
s^{(r+1)}_{\mu,\ell,1} &= (1-\rho_r)s^{(r)}_{\mu,\ell,1} + \rho_r\kappa^{-1}\sum_{d_i=\ell} p^{(r)}_{i,i}, \\
s^{(r+1)}_{\mu,\ell,2} &= T, \\
s^{(r+1)}_{\alpha,k,\ell,1} &= (1-\rho_r)s^{(r)}_{\alpha,k,\ell,1} + \rho_r\kappa^{-1}\sum_{d_i=\ell}\sum_{\substack{d_j=k\\ j<i}} p^{(r)}_{i,j}, \\
s^{(r+1)}_{\alpha,k,\ell,2} &= (1-\rho_r)s^{(r)}_{\alpha,k,\ell,1} + \kappa^{-1}\left(n^{(r)}_j - \sum_{d_j=k}\exp\left(-\beta^{(r)}_{k,l}\left(\kappa T - t_j\right)\right)\right), \\
s^{(r+1)}_{\beta,k,\ell,1} &= (1-\rho_r)s^{(r)}_{\beta,k,\ell,1} + \rho_r\kappa^{-1}\sum_{d_i=l}\sum_{\substack{d_j=k\\ j<i}} p^{(r)}_{i,j} \\
s^{(r+1)}_{\beta,k,\ell,2} &= (1-\rho_r)s^{(r)}_{\beta,k,\ell,2} + \rho_r\kappa^{-1}\sum_{d_i=l}\sum_{\substack{d_j=k\\ j<i}} p^{(r)}_{i,j}\left(t_i - t_j\right),
\end{aligned}
$$

where $n^{(r)}_j$ denotes the number of events on dimension j in $\mathbf{X}^{(r)}$. Finally, in the M-step, the value of the parameters is updated as:

$$
\alpha^{(r+1)}_{k,\ell} = \frac{s^{(r+1)}_{\alpha,k,\ell,1} + e_{k,\ell} - 1}{s^{(r+1)}_{\alpha,k,\ell,2} + f_{k,\ell}}, \qquad \beta^{(r+1)}_{k,\ell} = \frac{s^{(r+1)}_{\beta,k,\ell,1} + w_{k,\ell} - 1}{s^{(r+1)}_{\beta,k,\ell,2} + s_{k,\ell}}, \qquad \mu^{(r)}_{\ell} = \frac{s^{(r+1)}_{\mu,\ell,1} + a_\ell - 1}{s^{(r+1)}_{\mu,\ell,2} + b_\ell}.
$$

We repeat the steps above until the convergence criterion is reached.

3.3.3 Stochastic Gradient Variational Inference

Variational inference [158] is an approximate inference method that replaces the posterior distribution with an approximation that belongs to a tractable class. More specifically, the variational approximation $q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \mathbf{B})$, $\boldsymbol{\eta} \in H$ to the posterior distribution $p(\boldsymbol{\theta}, \mathbf{B} \mid \mathbf{X})$ is obtained through maximizing the evidence lower bound (ELBO), which is equivalent to setting

$$
\boldsymbol{\eta} = \arg\max_{\tilde{\boldsymbol{\eta}} \in H} \mathrm{E}_{q_{\tilde{\boldsymbol{\eta}}}} \log\left\{\frac{p(\boldsymbol{\theta}, \mathbf{B}, \mathbf{X})}{q_{\tilde{\boldsymbol{\eta}}}(\boldsymbol{\theta}, \mathbf{B})}\right\}. \tag{3.13}
$$

The class of variational approximations most used in practice is the class of mean-field approximations [14], where model parameters are taken to be independent from each other

under the variational distribution, i.e., $q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \mathbf{B}) = \prod_j q_{\boldsymbol{\eta}_{\theta_j}}(\theta_j) \prod_i q_{\boldsymbol{\eta}_{\mathbf{B}_i}}(\mathbf{B}_i)$. If both the full conditional posterior distributions and the corresponding variational distribution belong to the same exponential family, e.g., if

$$p\left(\theta_j \mid \boldsymbol{\theta}_{-j}, \mathbf{B}, \mathbf{X}\right) = A\left(\theta_j\right) \exp\left\{\theta_j s_j\left(\boldsymbol{\theta}_{-j}, \mathbf{B}, \mathbf{X}\right) - \psi\left(\boldsymbol{\theta}_{-j}, \mathbf{X}\right)\right\}$$
$$q_{\boldsymbol{\eta}_{\theta_j}}(\theta_j) = A(\theta_j) \exp\left\{\theta_j s_i\left(\boldsymbol{\eta}_{\boldsymbol{\theta}_j}\right) - \psi\left(\boldsymbol{\eta}_{\boldsymbol{\theta}_j}\right)\right\},$$

[16] showed that the coordinate ascent algorithm for the mean-field variational inference updates the variational parameters by setting $\boldsymbol{\eta}_{\theta_j}^{(r+1)} = \mathrm{E}_{q_{\boldsymbol{\eta}^{(\boldsymbol{r})}}}\left[s_j\left(\boldsymbol{\theta}_{-j}, \mathbf{B}, \mathbf{X}\right)\right]$. A similar result applies to the updates of the variational parameters $\boldsymbol{\eta}_{B_i}$.

Stochastic gradient variational inference (SGVI) [70] is a variant of variational inference that replaces the gradient computed over the whole sample with the one calculated over a random subsample $\mathbf{X}^{(r)}$ of size n selected during iteration r. Under conjugacy, SGVI then updates the vector $\boldsymbol{\eta}_{\mathbf{B}}$ in iteration r by setting

$$\eta_{\mathbf{B}_i^{(r)}}^{(r+1)} = \mathrm{E}_{q_{\eta^{(r)}}}\left[\tilde{s}_i\left(\mathbf{B}_{-i}^{(r)}, \boldsymbol{\theta}, \mathbf{X}^{(r)}\right)\right],$$

where $\tilde{s}_i\left(\mathbf{B}_{-i}^{(r)}, \boldsymbol{\theta}, \mathbf{X}^{(r)}\right)$ is the sufficient statistics associated with the block $\mathbf{B}_i$, and $\boldsymbol{\eta}_{\boldsymbol{\theta}}$ through the recursion

$$\eta_{\boldsymbol{\theta}_j}^{(r+1)} = (1 - \rho_r)\eta_{\boldsymbol{\theta}_j}^{(r)} + \rho_r \hat{\eta}_{\boldsymbol{\theta}_j}^{(r+1)},$$

where $\hat{\eta}_{\boldsymbol{\theta}_j}^{(r+1)} = \mathrm{E}_{q_{\eta^{(r+1)}}}\left[s_j(\boldsymbol{\theta}_{-j}, \mathbf{B}^{(r)}, \mathbf{X}^{(r)})\right]$. In the specific case of the MHP with exponential excitation functions we have $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, $\boldsymbol{\eta} = \left(\boldsymbol{\eta}_{\boldsymbol{\alpha}}, \boldsymbol{\eta}_{\boldsymbol{\beta}}, \boldsymbol{\eta}_{\boldsymbol{\mu}}, \boldsymbol{\eta}_{\boldsymbol{B}}\right)$ and

$$q_{\boldsymbol{\eta}}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, \mathbf{B}) = \prod_{i=1}^{N} q_{\boldsymbol{\eta}_{\mathbf{B}_i}}(\mathbf{B}_i) \prod_{k=1}^{K} q_{\boldsymbol{\eta}_{\boldsymbol{\mu}_k}}(\mu_k) \prod_{j=1}^{K} \prod_{k=1}^{K} q_{\boldsymbol{\eta}_{\boldsymbol{\alpha}_{k,\ell}}}(\alpha_{k,\ell}) q_{\boldsymbol{\eta}_{\boldsymbol{\beta}_{k,\ell}}}(\beta_{k,\ell}),$$

where $\alpha_{k,\ell} \sim \mathrm{Gamma}(\eta_{\alpha,k,\ell,1}, \eta_{\alpha,k,\ell,2})$, $\beta_{k,\ell} \sim \mathrm{Gamma}(\eta_{\beta,k,\ell,1}, \eta_{\beta,k,\ell,2})$, $\mu_\ell \sim \mathrm{Gamma}(\eta_{\mu,\ell,1}, \eta_{\mu,\ell,2})$, $\mathbf{B}_i$ denotes the i-th row of the matrix $\mathbf{B}$, and $\mathbf{B}_i$ follows a categorical distribution with parameter $\boldsymbol{\eta}_{\mathbf{B}_i}$. Hence, each iteration of the SGVI algorithm starts by updating the variational parameter for the local branching structure through the following formula:

$$\eta_{B_{ij}^{(r)}} \propto \begin{cases} \exp\left\{\psi\left(\eta_{\mu,d_i,1}^{(r)}\right) - \log\left(\eta_{\mu,d_i,2}^{(r)}\right)\right\} & j = i \\ \exp\left\{\Psi_{ij} - \log\left(\eta_{\alpha,d_j,d_i,2}^{(r)}\right) - \log\left(\eta_{\beta,d_j,d_i,2}^{(r)}\right)\right\} & j < i, \\ 0 & j > i, \end{cases}$$

where $\Psi_{ij} = \psi\left(\eta^{(r)}_{\alpha,d_j,d_i,1}\right) + \psi\left(\eta^{(r)}_{\beta,d_j,d_i,1}\right) - \frac{\eta^{(r)}_{\beta,d_j,d_i,1}}{\eta^{(r)}_{\beta,d_j,d_i,2}}\left(t_i^{(r)} - t_j^{(r)}\right)$. In this expression, $\psi(x) = \frac{\mathrm{d}}{\mathrm{d}x}\ln\Gamma(x)$ denotes the digamma function, and $(t_i^{(r)}, t_j^{(r)})$ represents the i-th and j-th event in $\mathbf{X}^{(r)}$. Then, we update the rest of the variational parameters as:

$$\eta^{(r+1)}_{\alpha_{k,\ell,1}} = (1-\rho_r)\eta^{(r)}_{\alpha_{k,\ell,1}} + \rho_r\left(\kappa^{-1}\sum_{d_i=\ell}\sum_{\substack{d_j=k\\ j<i}}\eta_{B^{(r)}_{ij}} + e_{k,\ell}\right),$$

$$\eta^{(r+1)}_{\alpha_{k,\ell,2}} = (1-\rho_r)\eta^{(r)}_{\alpha_{k,\ell,2}} + \rho_r\left(\kappa^{-1}\left(n_k^{(r)} - \sum_{d_j=k}\left(1+\frac{\kappa T - t_j}{\eta_{\beta^{(r+1)}_{k,\ell,2}}}\right)^{-\eta^{(r+1)}_{\beta_{k,l,1}}}\right) + f_{k,\ell}\right),$$

$$\eta^{(r+1)}_{\beta_{k,\ell,1}} = (1-\rho_r)\eta^{(r)}_{\beta_{k,\ell,1}} + \rho_r\left(\kappa^{-1}\sum_{d_i=\ell}\sum_{\substack{d_j=k\\ j<i}}\eta_{B^{(r)}_{ij}} + r_{k,\ell}\right),$$

$$\eta^{(r+1)}_{\beta_{k,\ell,2}} = (1-\rho_r)\eta^{(r)}_{\beta_{k,\ell,2}} + \rho_r\left(\kappa^{-1}\sum_{d_i=\ell}\sum_{\substack{d_j=k\\ j<i}}\eta_{B^{(r)}_{ij}}(t_i^{(r)} - t_j^{(r)}) + s_{k,\ell}\right),$$

$$\eta^{(r+1)}_{\mu_{\ell,1}} = (1-\rho_r)\eta^{(r)}_{\mu_{\ell,1}} + \rho_r\left(\kappa^{-1}\sum_{d_i=\ell}\eta_{B^{(r)}_{ii}} + a_\ell\right),$$

$$\eta^{(r+1)}_{\mu_{\ell,2}} = T + b_\ell.$$

These updates are repeated until convergence.

3.3.4 Stochastic Gradient Langevin Dynamics

Unlike the previous two sections, here we focus on inference methods that are based on the observed data likelihood (3.2) instead of the complete data likelihood (3.3). Specifically, we consider simulation methods that rely on Langevin dynamics (LD) [118], a class of MCMC methods that are based on the discretization of a continuous-time stochastic process whose equilibrium distribution is the desired posterior distribution. Compared to simple random walk MCMC algorithms, LD algorithms explore the parameter space much more efficiently because they use information about the gradient of the likelihood to guide the direction of the random walk. In particular, LD methods proposes new values for the parameter

according to

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(r)} - \frac{\rho}{2} \nabla_{\boldsymbol{\theta}} U\left(\boldsymbol{\theta} \mid \mathbf{X}\right)\big|_{\theta=\boldsymbol{\theta}^{(r)}} + \sqrt{\rho}\epsilon_{r+1}, \tag{3.14}$$

where ρ is the step size used to discretize the Langevin diffusion, $U\left(\boldsymbol{\theta} \mid \mathbf{X}\right) = -\log p(\mathbf{X} \mid \boldsymbol{\theta}) - \log p(\boldsymbol{\theta})$ is the negative logarithm of the unnormalized posterior of interest, and ϵ_{r+1} is drawn from a standard multivariate normal distribution. If no discretization of the Langevin diffusion was involved, then this proposed valued would come from the correct stationary distribution. However, the introduction of the discretization means that a correction is required. Hence, values proposed according to (3.14) are accepted with probability

$$\min\left\{1, \frac{\exp\left\{-U(\boldsymbol{\theta}^* \mid \mathbf{X})\right\}}{\exp\left\{-U(\boldsymbol{\theta}^{(r)} \mid \mathbf{X})\right\}}\right\}. \tag{3.15}$$

If accepted, then $\boldsymbol{\theta}^{(r+1)} = \boldsymbol{\theta}^*$. Otherwise, $\boldsymbol{\theta}^{(r+1)} = \boldsymbol{\theta}^{(r)}$.

The stochastic gradient Langevin Dynamics (SGLD) algorithm [162, 28] replaces the likelihood computed over the whole sample with (an appropriately rescaled) likelihood evaluated on a random subsample $\mathbf{X}^{(r)}$. SGLD also uses a decreasing stepsize ρ_r to construct the discretization of the Langevin diffusion in step r of the algorithm and ignores the correction step in (3.15). This leads to updates of the form

$$\boldsymbol{\theta}^{(r+1)} = \boldsymbol{\theta}^{(r)} - \frac{\rho_r}{2} \nabla_{\boldsymbol{\theta}} \tilde{U}(\boldsymbol{\theta} \mid \mathbf{X}^{(r)})\Big|_{\theta=\boldsymbol{\theta}^{(r)}} + \sqrt{\rho_r}\epsilon_{r+1}, \tag{3.16}$$

where $\tilde{U}(\boldsymbol{\theta} \mid \mathbf{X}^{(r)}) = \kappa^{-1} \log p\left(\mathbf{X}^{(r)} \mid \boldsymbol{\theta}\right) + \log p\left(\boldsymbol{\theta}\right)$.

In the case of the MHP model with exponential excitation functions, we perform a logarithmic transformation on the model parameters before implementing the SGLD, so

that $\boldsymbol{\xi_\alpha} = \log \boldsymbol{\alpha}$, $\boldsymbol{\xi_\beta} = \log \boldsymbol{\beta}$ and $\boldsymbol{\xi_\mu} = \log \boldsymbol{\mu}$. Then, the gradients become:

$$\begin{aligned}
\nabla^{(r)}_{\xi_{\alpha_{k,\ell}}} U(\boldsymbol{\xi}) &= -\sum_{d_i=\ell} \frac{\alpha^{(r)}_{k,\ell}\beta^{(r)}_{k,\ell}\sum_{d_j=k,j<i}\exp\left(-\beta^{(r)}_{k,\ell}\left(t^{(r)}_i - t^{(r)}_j\right)\right)}{\mu^{(r)}_\ell + \alpha^{(r)}_{k,\ell}\beta^{(r)}_{k,\ell}\sum_{d_j=k,j<i}\exp\left(-\beta^{(r)}_{k,\ell}\left(t^{(r)}_i - t^{(r)}_j\right)\right)} \\
&\quad + \alpha^{(r)}_{k,\ell}\left(n^{(r)}_k - \sum_{d_j=k}\exp\left(-\beta^{(r)}_{k,l}\left(\kappa T - t_j\right)\right) + f_{k,\ell}\right) - e_{k,\ell}, \\
\nabla^{(r)}_{\xi_{\beta_{k,\ell}}} U(\boldsymbol{\xi}) &= -\sum_{d_i=\ell} \frac{\alpha^{(r)}_{k,\ell}\beta^{(r)}_{k,\ell}\sum_{d_j=k,j<i}\left(1-\beta^{(r)}_{k,\ell}\left(t^{(r)}_i - t^{(r)}_j\right)\right)\exp\left(-\beta^{(r)}_{k,\ell}\left(t^{(r)}_i - t^{(r)}_j\right)\right)}{\mu^{(r)}_\ell + \alpha^{(r)}_{k,\ell}\beta^{(r)}_{k,\ell}\sum_{d_j=k,j<i}\exp\left(-\beta^{(r)}_{k,\ell}\left(t^{(r)}_i - t^{(r)}_j\right)\right)} \\
&\quad + \sum_{d_j=k}\alpha^{(r)}_{k,l}(\kappa T - t_j)\exp\left(-\beta^{(r)}_{k,l}\left(\kappa T - t_j\right)\right) - r_{k,\ell} + s_{k,\ell}\beta^{(r)}_{k,\ell}, \\
\nabla^{(r)}_{\xi_{\mu_{k,\ell}}} U(\boldsymbol{\xi}) &= -\sum_{d_i=\ell} \frac{\mu^{(r)}_\ell}{\mu^{(r)}_\ell + \alpha^{(r)}_{k,\ell}\beta^{(r)}_{k,\ell}\sum_{d_j=k,j<i}\exp\left(-\beta^{(r)}_{k,l}\left(t^{(r)}_i - t^{(r)}_j\right)\right)} + \mu^{(r)}_\ell(b_\ell + \kappa T) - a_\ell.
\end{aligned}$$

Note that SGLD does not require approximating the observed data likelihood.

3.4 Simulation studies

In this section, we conduct a series of simulations to understand the performance of the algorithms with and without time budget constraints. Compared with small-scale learning problems, large-scale problems are subject to a qualitatively different tradeoffs involving the computational complexity of the underlying algorithm [20], making evaluation under time constraints key. We also investigate the model fitting performance of all algorithms under different subsampling ratios.

3.4.1 Experimental setting

Data generation mechanism. For most of our experiments, data is generated from the multivariate Hawkes process model presented in section 3.2 with $K = 3$ dimensions and the following parameter settings:

$$\boldsymbol{\alpha} = \begin{bmatrix} 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \end{bmatrix}, \qquad \boldsymbol{\beta} = \begin{bmatrix} 4 & 4 & 4 \\ 4 & 4 & 4 \\ 4 & 4 & 4 \end{bmatrix}, \qquad \boldsymbol{\mu} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}.$$

In addition, for our last set of experiments, we also consider an alternative data generation mechanism for a 10-dimensional Hawkes processes with varying degrees of sparsity on the matrix $\boldsymbol{\alpha}$. More specifically, we let have $\mu_\ell = 0.1, \beta_{k,\ell} = 4$ for all $k, \ell = 1, \ldots, 10$ and consider three scenarios ('high', 'medium' and 'low' sparsity), under which 90%, 80% and 50% of the off-diagonal elements of $\boldsymbol{\alpha}$ are set to zero, indicating no interaction between the dimension pairs. The remaining off-diagonal elements of $\boldsymbol{\alpha}$ are set to the value 0.1, and the diagonal elements to the value 0.4.

Algorithms to be compared. We compare the performances of SGLD, versions of SGEM, SGVI that use the standard approximation of [90], and the boundary-corrected versions of SGEM and SGVI (SGEM-c and SGVI-c). Also, as a 'gold-standard' that does not involve subsampling, we implemented full MCMC and its boundary-corrected version (MCMC-c). Additionally, we benchmark our methods against two frequentist computational methods for MHP: the nonparametric estimation based on EM algorithm and piecewise basis kernels (EM-BK, see 170) and the maximum likelihood estimation for multidimensional exponential Hawkes process with both excitation and inhibition effects (MLE-I, see 18).

Parameters. For the model hyperparameters from Section 3.2.2, we let $a_\ell = 2, b_\ell = 4, e_{k,\ell} = 2, f_{k,\ell} = 4, r_{k,\ell} = 2, s_{k,\ell} = 0.5$ for $k, \ell = 1, \ldots, K$. We simulate $K_d = 50$ datasets for $T = 1000$. For every dataset, we start all algorithms at 16 different initial points to minimize the risk of convergence to a local optimum. For the tuning hyperparameters in stochastic optimization algorithms, we consider several subsampling ratios of $\kappa = \{0.01, 0.05, 0.1, 0.2, 0.3\}$ and let $\tau = 1, \kappa = 0.51$. For SGEM and SGVI, we let $\rho_0 = 0.02$, and for SGLD we let $\rho_0 = \frac{0.1}{T\kappa}$. We chose $\delta = 0.25$ as the threshold for boundary-corrected methods.

Performance Metrics for Model Fitting. We consider the observed data likelihood defined in (3.2) as a measure for model fitting. Denote by $\mathrm{ODL}_{d,\iota}$ the observed data likelihood calculated based on dataset d and initial point ι, we define $\mathrm{BODL}_d = \max_{1 \le \iota \le 16} \mathrm{ODL}_{d,\iota}$ as the best-observed data likelihood (BODL), as a basis for evaluating model performance. Finally, in order to compare model-fitting performance under different subsampling ratios and

different datasets, we propose the following relative best-observed data likelihood (RBODL):

$$\text{RBODL}_{d,\kappa_1,\kappa_2} = \frac{\text{BODL}_{d,\kappa_1}}{\text{BODL}_{d,\kappa_2}}$$

where $\text{BODL}_{d,\kappa_1}, \text{BODL}_{d,\kappa_2}$ are the best-observed data likelihoods on dataset d under subsampling ratio κ_1 and κ_2. Additionally, we refer to κ_2 as the reference subsampling ratio for $\text{RBODL}_{d,\kappa_1,\kappa_2}$. The RBODL is fairly easy to interpret, in that $\text{RBODL}_{d,\kappa_1,\kappa_2} > 1$ indicates a superior empirical performance of subsampling ratio κ_1 compared to κ_2 and vice versa.

Performance Metrics for Estimation Accuracy. We consider performance metrics for both point and uncertainty estimations. To evaluate estimation accuracy of the model parameters, we rely on the averaged root mean integrated squared error (RMISE) for $\boldsymbol{\alpha}, \boldsymbol{\beta}$, and use mean absolute error (MAE) for $\boldsymbol{\mu}$ on the log scale:

$$\text{RMISE}(\boldsymbol{\alpha}, \boldsymbol{\beta}) := \frac{1}{K^2} \sum_{j=1}^{K} \sum_{k=1}^{K} \sqrt{\int_0^{+\infty} \left(\phi_{j,k}^{\text{true}}(x) - \hat{\phi}_{j,k}(x) \right)^2 \, \mathrm{d}x},$$

$$\text{MAE}(\boldsymbol{\mu}) := \frac{1}{K} \sum_{j=1}^{K} |\log(\mu_k^{\text{true}}) - \log(\hat{\mu}_k)|.$$

where $\hat{\mu}_k$ is the point estimator of μ_k (the posterior mode for the stochastic gradient EM, the posterior mean under the variational approximation for the stochastic gradient variational method, and the posterior mean of the samples after burn-in for the stochastic gradient Langevin dynamics), and $\hat{\phi}_{k,\ell}(x)$ is obtained by plugging in the point estimators for $\alpha_{k,\ell}$ and $\beta_{k,\ell}$ into the exponential decay function. The RMISE is a commonly used metric for nonparametric triggering kernel estimation for MHP models [169] and collectively evaluates the estimation performance for all model parameters.

We also evaluate the uncertainty estimates generated by the SGVI, SGLD, SGVI-c and SGLD-c models (SGEM provides point estimators, but does not directly provide estimates of the posterior variance). To do so, we consider the interval score (IS) [59] for 95% credible intervals, which jointly evaluates the credible interval width and its coverage rate. We also separately compute the average coverage rate (ACR), defined as the proportion of correct coverages out of $2K^2 + K$ model parameters and the average credible interval length (AIW) as references.

3.4.2 Simulation results

methods	running time	0.05	0.1	0.2	0.3
SGEM	1 min	**1.003** (0.003)	1.002 (0.004)	1.000 (0.003)	0.999 (0.004)
	3 min	**1.004** (0.005)	1.003 (0.005)	1.002 (0.005)	1.001 (0.006)
	5 min	1.003 (0.005)	**1.004** (0.006)	1.002 (0.006)	1.002 (0.006)
SGEM-c	1 min	**1.003** (0.007)	1.002 (0.006)	1.000 (0.006)	0.999 (0.006)
	3 min	**1.003** (0.005)	1.003 (0.006)	1.002 (0.006)	1.001 (0.006)
	5 min	1.003 (0.008)	**1.004** (0.008)	1.003 (0.007)	1.003 (0.008)
SGVI	1 min	**1.004** (0.001)	1.002 (0.001)	1.000 (0.001)	0.997 (0.001)
	3 min	**1.005** (0.001)	1.004 (0.001)	1.002 (0.001)	1.001 (0.001)
	5 min	**1.005** (0.001)	1.005 (0.001)	1.003 (0.001)	1.002 (0.001)
SGVI-c	1 min	**1.002** (0.001)	1.000 (0.001)	0.997 (0.001)	0.995 (0.001)
	3 min	**1.002** (<0.001)	1.002 (0.001)	1.000 (0.001)	0.998 (0.001)
	5 min	**1.002** (<0.001)	1.002 (0.001)	1.001 (0.001)	0.999 (0.001)
SGLD	1 min	**1.001** (<0.001)	0.996 (0.001)	0.988 (0.002)	0.98 (0.004)
	3 min	**1.001** (<0.001)	0.998 (0.001)	0.991 (0.001)	0.986 (0.003)
	5 min	**1.001** (<0.001)	0.999 (0.001)	0.992 (0.001)	0.987 (0.002)

Table 3.1: RBODLs for SGEM, SGVI and SGLD under running times of 1, 3 and 5 minutes for the first data generation mechanism ($K = 3$), with $\kappa = 0.01$ being the reference subsampling ratio. Average RBODL across 50 datasets is shown, with standard deviations in the brackets.

Optimal subsampling ratios. Table 3.1 shows the RBODLs for all methods subject to three time budgets: 1, 3 and 5 minutes. We choose $\kappa = 0.01$ as the reference subsampling ratio. The results indicate that, except for SGLD run for 5 minutes, all methods reach the highest RBODL at $\kappa = 0.05$. Given that this optimun is greater than 1, this indicates that

choosing a subsampling ratio around 0.05 (rather than the baseline, 0.01) leads to optimal model-fitting performance under a time budget. For a given method under fixed running time, we observe that the RBODL tends to drop as κ increases. This is likely because larger subsamples take considerably more time to process due to the quadratic computational complexity for each iteration. We also observe such drops in RBODL tend to reduce in size as running times increase, which suggests better model convergence with more computation time. Finally, we see more dramatic drops in RBODL for SGVI compared to SGEM under the same running time, which suggests that the EM algorithms tends to converge faster than VI algorithms. This result concurs with those of [169] in the non-stochastic gradient setting.

methods	RMISE ($\boldsymbol{\alpha}, \boldsymbol{\beta}$)	MAE ($\boldsymbol{\mu}$)	IS	ACR	AIW
MCMC	0.042	0.072	1.042	0.952	1.015
	(0.008)	(0.036)	(0.274)	(0.046)	(0.053)
MCMC-c	0.042	0.072	1.056	0.952	1.015
	(0.008)	(0.036)	(0.278)	(0.055)	(0.058)
SGLD	0.052	0.109	3.898	0.667	0.844
	(0.019)	(0.283)	(4.739)	(0.152)	(0.172)
SGVI	0.046	0.103	6.163	0.333	0.222
	(0.008)	(0.048)	(2.117)	(0.131)	(0.012)
SGVI-c	0.040	0.093	4.905	0.429	0.213
	(0.007)	(0.044)	(1.698)	(0.139)	(0.012)
SGEM	0.100	0.024	-	-	-
	(0.076)	(0.026)			
SGEM-c	0.103	0.023	-	-	-
	(0.065)	(0.022)			
MLE-I	0.030	0.077	-	-	-
	(0.007)	(0.036)			
EM-BK	0.340	2.065	-	-	-
	(0.009)	(0.256)			

Table 3.2: Estimation metrics across all nine methods for the first data generation mechanism ($K = 3$). The values in the grid cells are the average across 50 datasets, with the standard deviation in the brackets.

Estimation accuracy. Table 3.2 shows the estimation performance measures, including RMISE, MAE, IS, ACR and AIW for all seven methods, along with the two frequentist benchmark methods. Similar to the previous simulation study, we run the same algorithm on 50 datasets with 16 different initial parameter values and choose the instance associated

with the highest observed data likelihood for estimation performance evaluation. We keep the same stochastic optimization hyperparameters, fixing the subsampling ratio κ at 0.05. For SGLD, SGVI, SGVI-c, SGEM and SGEM-c, we run the algorithms for 30 minutes. For MCMC and MCMC-c, we run the algorithms on the whole dataset without subsampling for 15,000 iterations, which took around 12 hours to complete. We discard the first 5,000 samples as burn-in and calculate the posterior median for estimation performance evaluation. For both optimization-based frequentist methods, we run the algorithms until they have converged under the default convergence criteria. As would be expected, the MCMC algorithms have values of RMISE, MAE and IS lower than the majority of other methods. Moreover, MCMC algorithms produce coverage rates that are very close to nominal. Among the remaining methods from Section 3, SGLD shows the best uncertainty estimation performance with the lowest IS and ACR closest to the nominal rate, while SGVI-c shows the best point estimation performance with RMISE even lower than the MCMC methods. Additionally, we observe a significant improvement in both RMISE and IS for SGVI-c compared to SGVI, indicating that incorporating a boundary correction can lead to both improved point and uncertainty estimation performance for SGVI. For the frequentist benchmark methods, MLE-I has the lowest RMISE among all nine methods in this simulation scenario, while having MAE higher than most methods. EM-BK has the worst point estimation accuracy across the board.

Sensitivity analysis for different dataset sizes. We also look at how sensitive the RBODLs are to the scale of the data we have. We run the same algorithms on two sets of 50 datasets with $T = 500$ and $T = 2000$, and the RBODLs are shown in Tables 3.6 and 3.7. For the small datasets, the median RBODLs change much less than the large datasets over different subsampling ratios. This is not surprising, as it indicates that the algorithms tend to reach convergence faster for smaller datasets than the larger ones. Additionally, the optimal subsampling ratios for smaller datasets tend to be larger, indicating that there could be a fixed amount of data needed for algorithms to attain better model-fitting results.

Sensitivity analysis for the stochastic search parameters. Next, we investigate the effect of the stochastic search parameters on our previous simulation results. To this end, we rerun our analysis for the medium-sized dataset under each of the following three sets of τ_1, τ_2 values: (1) $\tau_1 = 5, \tau_2 = 0.51$, (2) $\tau_1 = 1, \tau_2 = 1$, (3) $\tau_1 = 5, \tau_2 = 1$. The results are shown in Table 3.8, 3.9 and 3.10. As expected, the behavior of RBODL with respect to subsampling ratio and running time is similar to the default scenario shown in Table 3.1. Also, the results in Table 3.8 are more similar to those in Table 3.1 than cases in Tables 3.9 and 3.10. This is because τ_2 controls the decay rate of the stepsize parameter, which has a bigger long-term effect on ρ_r compared to the delay parameter τ_1. We also looked at the estimation performances for all five methods under these four scenarios, with performance metrics shown in Tables 3.11, 3.12, 3.13 and 3.14. We can see that all algorithms performed significantly better in scenarios where $\tau_2 = 0.51$, indicating that a large value of τ_2 may lead to suboptimal estimates because algorithms converged too fast. We also note that the SGEM-c outperformed SGEM where $\tau_2 = 0.51$.

Sensitivity analysis for the threshold values of the boundary-corrected methods. Previously, we chose a fixed value $\delta = 0.25$ as the common threshold for all boundary-corrected methods. In this simulation study, we investigate a systematic way of choosing δ based on the discussion in Section 3.2.1, and study the parameter estimation performance under different values of δ. Given an estimate for $\boldsymbol{\beta}$ and a fixed value $r > 0$, we find δ such that $\delta = \frac{1}{K^2} \sum_{j=1}^{K} \sum_{k=1}^{K} \frac{1}{\hat{\beta}_{k,\ell}}$. Table 3.15 shows the point and uncertainty estimation results for SGVI-c and SGEM-c under values of $r \in \{0.5, 1, 2, 3, 4\}$. For both methods, all estimation metrics reached optimality between $r = 1$ and $r = 2$.

methods	RMISE $(\boldsymbol{\alpha}, \boldsymbol{\beta})$	MAE $(\boldsymbol{\mu})$	IS	ACR	AIW
MCMC-rw	0.044	0.075	1.242	0.951	1.037
	(0.008)	(0.037)	(0.418)	(0.057)	(0.063)
SGLD-apx	0.050	0.113	2.494	0.763	0.876
	(0.012)	(0.047)	(1.483)	(0.111)	(0.120)

Table 3.3: Estimation metrics for MCMC-rw and SGLD-apx for the first data generation mechanism ($K = 3$). The values in the grid cells are the average across 50 datasets, with the standard deviation in the brackets.

Sensitivity analysis for boundary approximation. To numerically show that the boundary approximations in (3.5) and (3.6) does not lead to a significant information loss, we implemented two additional methods as a comparison: the full MCMC algorithm with random-walk updates for $\boldsymbol{\beta}$ using the full likelihood (MCMC-rw), and the Langevin dynamics algorithm with likelihood approximation (SGLD-apx). We focus on these two algorithms here because they are ones where both the true and the various approximate likelihoods can be implemented in a straightforward fashion. Table 3.3 shows the results for these two methods, which should be compared with those for MCMC, MCMC-c and SGLD in Table 3.11. The results suggest that the errors introduced by the approximation are negligible (at least, compared to the Monte Carlo error involved), both in terms of both point and uncertainty estimation.

Robustness to model sparsity. Our last evaluation uses the alternative data generation mechanism described in Section 3.4.1 where we have a ten-dimensional MHP with varying degrees of sparsity in $\boldsymbol{\alpha}$. Tables 3.4 and 3.5 show the average and standard deviation of the estimation metrics for all five methods and the two frequentist benchmarks. Due to presence of sparsity, the metrics for the $\alpha_{k,\ell}$s and $\beta_{k,\ell}$s are evaluated only for entries that are associated with a non-zero true value for $\alpha_{k,\ell}$. Note that, unlike the first simulation

scenario, MLE-I and EM-BK perform significantly worse than the Bayesian methods in all of our simulations with $K = 10$. Furthermore, MLE-I takes much longer time to converge compared to the three-dimensional scenario. For example, the average running time for the 'low sparsity' scenario is 61.47 minutes, which is much longer than the 30-minute running time for all five stochastic Bayesian methods. Aside from this, the results are largely consistent those from the original, three-dimensional data generation mechanism. SGVI and SGVI-c have the best point estimation performance, SGEM and SGEM-c have the worst, and SGLD is in between. Furthermore, scenarios that are less sparse tend to be associated with lower estimation errors.

methods	sparsity	RMISE ($\boldsymbol{\alpha}, \boldsymbol{\beta}$)	MAE ($\boldsymbol{\mu}$)	IS	ACR	AIW
	high	0.047 (0.007)	0.308 (0.033)	-	-	-
SGEM	medium	0.039 (0.005)	0.263 (0.030)	-	-	-
	low	0.030 (0.003)	0.160 (0.026)	-	-	-
	high	0.047 (0.006)	0.263 (0.034)	-	-	-
SGEM-c	medium	0.039 (0.005)	0.161 (0.030)	-	-	-
	low	0.030 (0.003)	0.161 (0.027)	-	-	-
	high	0.206 (0.015)	0.331 (0.151)	-	-	-
MLE-I	medium	0.155 (0.008)	0.421 (0.199)	-	-	-
	low	0.124 (0.003)	0.247 (0.195)	-	-	-
	high	0.462 (0.047)	0.584 (0.106)	-	-	-
EM-BK	medium	0.366 (0.048)	0.445 (0.106)	-	-	-
	low	0.261 (0.019)	0.197 (0.088)	-	-	-

Table 3.5: Estimation metrics across SGEM , SGEM-c, MLE-I and EM-BK for the second data generation mechanism ($K = 10$) with three levels of model sparsity. The values in the grid cells are the average across 50 datasets, with the standard deviation in the brackets.

methods	sparsity	RMISE $(\boldsymbol{\alpha}, \boldsymbol{\beta})$	MAE $(\boldsymbol{\mu})$	IS	ACR	AIW
SGLD	high	0.048 (0.005)	0.295 (0.035)	1.562 (0.546)	0.742 (0.033)	1.003 (0.085)
	medium	0.041 (0.005)	0.256 (0.037)	2.172 (0.680)	0.802 (0.036)	1.327 (0.144)
	low	0.031 (0.003)	0.171 (0.018)	2.915 (1.033)	0.841 (0.034)	1.443 (0.072)
SGVI	high	0.047 (0.013)	0.179 (0.038)	2.011 (0.735)	0.699 (0.059)	0.680 (0.112)
	medium	0.040 (0.008)	0.158 (0.039)	3.158 (1.131)	0.704 (0.053)	0.766 (0.105)
	low	0.030 (0.004)	0.111 (0.029)	5.264 (1.269)	0.620 (0.045)	0.664 (0.089)
SGVI-c	high	0.047 (0.012)	0.183 (0.038)	1.904 (0.729)	0.703 (0.062)	0.665 (0.047)
	medium	0.039 (0.008)	0.163 (0.039)	2.941 (1.051)	0.710 (0.055)	0.747 (0.051)
	low	0.029 (0.004)	0.118 (0.029)	4.515 (1.110)	0.637 (0.048)	0.638 (0.033)

Table 3.4: Estimation metrics across SGLD, SGVI and SGVI-c for the second data generation mechanism ($K = 10$) with three levels of model sparsity. The values in the grid cells are the average across 50 datasets, with the standard deviation in the brackets.

3.5 Real-world application

Data description. In this section, we apply our methods to model the market risk dynamics in the Standard & Poor (S&P)'s 500 intraday index prices for its 11 sectors: Consumer Discretionary (COND), Communication Staples (CONS), Energy (ENRS), Financials (FINL), Health Care (HLTH), Industrials (INDU), Information Technology (INFT), Materials (MATR), Real Estate (RLST), Communication Services (TELS), Utilities (UTIL). To achieve this, price data between August 22, 2022 and Jan 23, 2023 was downloaded from Bloomberg Finance L.P. Similar to [140], an event occurs on dimension $k = 1, \ldots, 11$ if the negative log returns in sector k exceeds a predetermined threshold (in our case, a 0.05% drop on a one-minute basis). The resulting dataset contains 55,509 events across the 11 dimensions.

Results. We fit a Hawkes process model with exponential decay functions to the event data using the SGEM, SGEM-c, SGVI, SGVI-c and SGLD algorithms. We set the subsampling ratio of $\kappa = 0.01$ for SGLD and of $\kappa = 0.05$ for all other methods. Similar to the procedure in Section 3.4.1, we start all algorithms at 16 different initial points and choose the instances with the highest observed data likelihood to compute the estimates. Furthermore, all these algorithms were run for a fixed period of 30 minutes for each initial set of values. As a reference, we also apply MCMC and MCMC-c to the dataset and use 10,000 posterior samples after 10,000 burn-ins, which roughly took around two days.

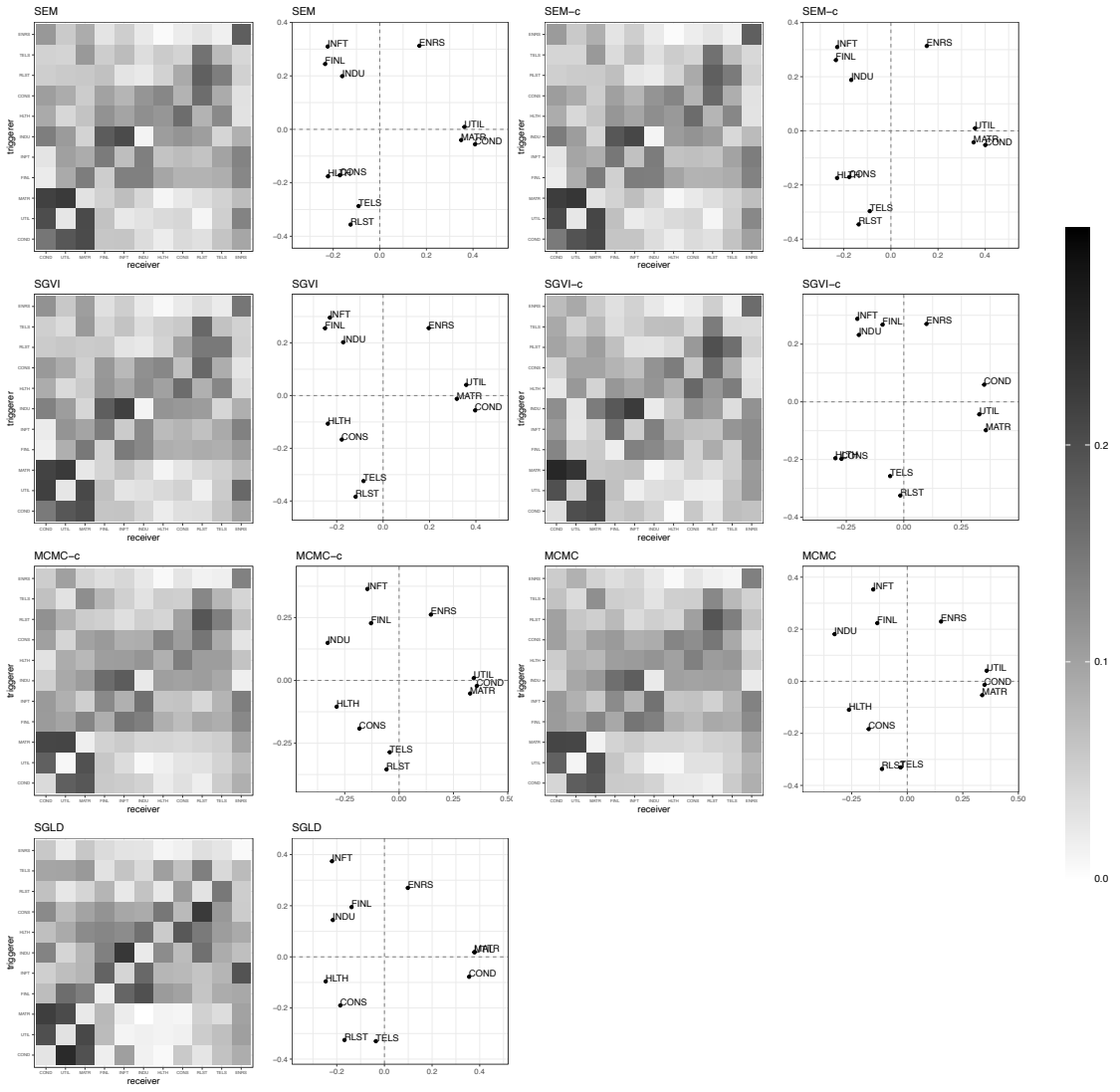

Figure 3.2: Left panel: heatmaps for point estimates of $\boldsymbol{\alpha}$ parameters for the corresponding 11 sectors, for all seven algorithms. Right panel: first two principal coordinates (after applying the Procrustes analysis algorithm) for the 11 sectors based on the distance measure matrix for $\boldsymbol{\alpha}$ estimates.

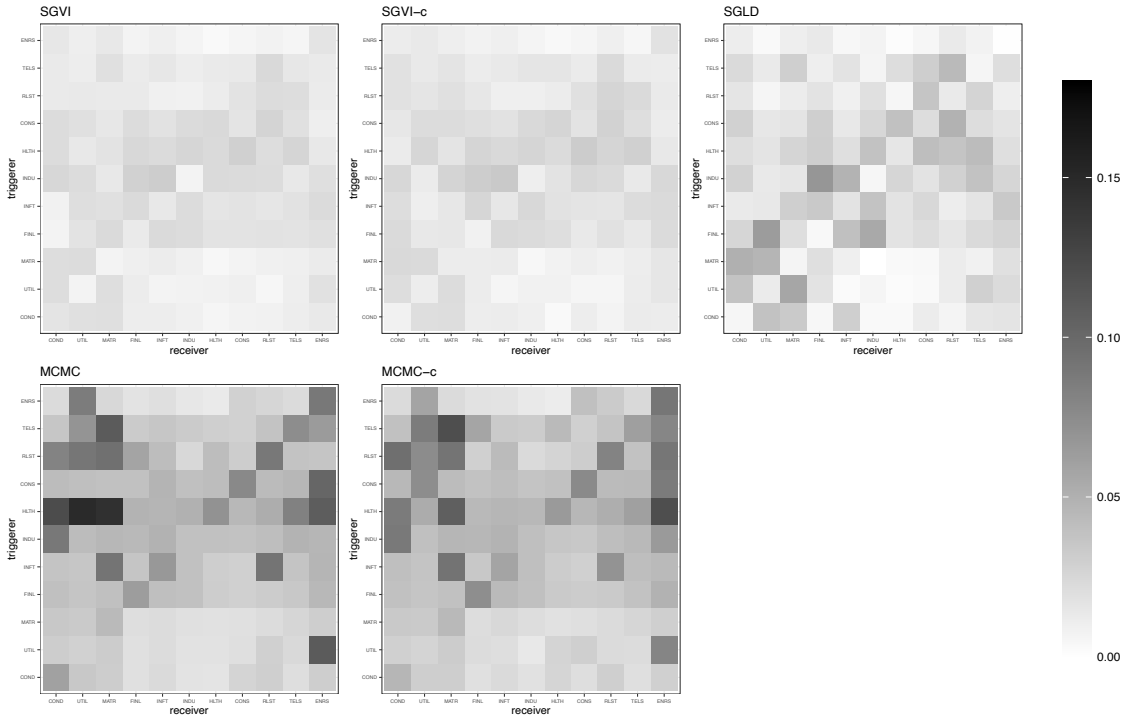

Figure 3.3: Heatmaps for 95% credible interval lengths estimates of $\boldsymbol{\alpha}$ parameters for the corresponding 11 sectors, for all five algorithms that yield uncertainty estimates.

Figure 3.2 shows heatmaps of point estimates for the $\boldsymbol{\alpha}$ parameters for all seven algorithms. To facilitate comparisons, we also generate a visual representation by constructing a measure of similarity between sectors i and j as $\Upsilon(i,j) = \exp\left\{-\frac{1}{2}(\alpha_{ij} + \alpha_{ji})\right\}$, and then use multidimensional scaling [154] to find a two-dimensional representation of these similarities. Because the representation is arbitrary up to translations, rotations and reflections, we use Procrustes analysis [43] to align the representations for all the algorithms.

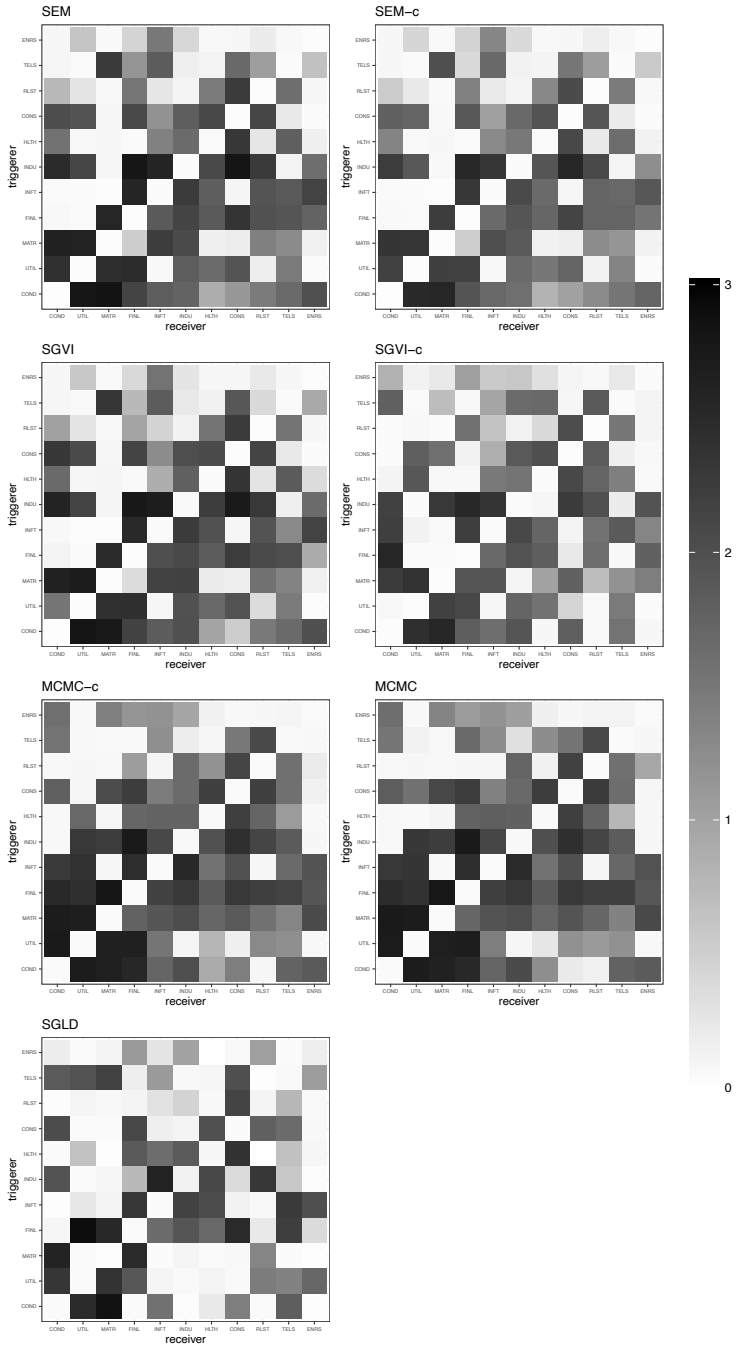

Figure 3.4: Heatmaps for point estimates of $\boldsymbol{\beta}$ parameters for the corresponding 11 sectors, for all seven algorithms.

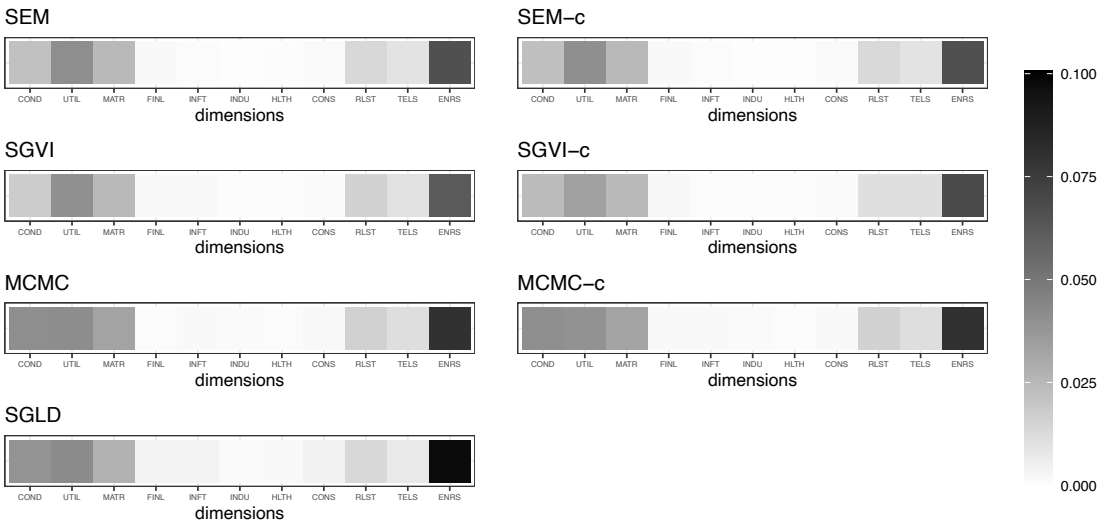

Figure 3.5: Heatmaps for point estimates of $\boldsymbol{\mu}$ parameters for the corresponding 11 sectors, for all seven algorithms.

All seven methods yield similar point estimates for $\boldsymbol{\alpha}$. To explore this question in more detail, we present in Table 3.16 the mean square distance between point estimates for each pair of methods. We see that MCMC and MCMC-c are almost identical, and that SGEM, SGVI, and SGEM-c yield very similar estimates. Interestingly, SGVI-c yields results that are as close to those of the MCMC "gold standard" as those from SGEM, SGVI, and SGEM-c, but that are fairly different from them. We also note that SGLD seems to yield results that are the furthest away from the MCMC procedures, suggesting that a time budget of 30 minutes is not enough to achieve reliable results in this example. From a substantive point of view, Figure 3.2 suggests mutual excitation of exceedances within each of the following three groups: (1) UTIL, MATR COND, (2) INFT, FINL and INDU, (3) TELS, RLST, HLTH and CONS and ENRS. One particular interesting result is the estimates for the energy (ENRS) sector, which has a much higher diagonal α estimate and lower off diagonal estimates corresponding to other sectors. This is supported by the scatterplot of principal coordinates, in which the point for ENRS is away from all other sectors, indicating that such sector may be less likely associated with price movements in other sectors.

Next, we show in Figure 3.4 point estimates of $\boldsymbol{\beta}$ under all seven methods and, in Table

3.17, the mean square distance between the estimates generated by the different methods. The pattern of the results is very similar: (1) MCMC and MCMC-c yield the most similar results, (2) SGLD seems to yield estimates that are furthest away from those generated by the MCMC methods, (3) SGEM, SGVI, and SGEM-c yield very similar results to each other, and (4) SGVI-c yields different results from SGEM, SGVI, and SGEM-c, but they are as close to those of the MCMC approaches as those from the three alternatives. We note, however, that the estimates of $\boldsymbol{\beta}$ generated by MCMC and MCMC-c do seem to differ from each other much more than than the estimates of $\boldsymbol{\alpha}$ did.

Figure 3.5 shows the point estimates for $\boldsymbol{\mu}$, and Table 3.18 shows mean square distances between the model estimates. Not surprisingly, the same patterns arise again, although we note that the distances tend to be smaller. From an application point of view, we note that all methods identify ENRS as a sector with a very high baseline rate events, FINL, INFT, INDU, HLTH and CONS as sectors where the majority of price drops are the result of contagion from turbulence in other sectors.

As for uncertainty estimation, Figures 3.3, 3.13 and 3.14 show the length of the estimated posterior credible intervals for $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ for SGVI, SGVI-c and SGLD, as well as for MCMC and MCMC-c. As was the case with simulated datasets, stochastic gradient methods seem to underestimate the uncertainty in the posterior distribution, with SGVI and SGVI-c doing much more dramatically than SGLD.

Goodness-of-fit analysis for the real-world example. Finally, we conduct in-sample goodness-of-fit analysis using quantile-quantile plots for the posterior distribution of inter-event times. The construction of these plots relies on the well-known time-rescaling theorem [33], which states that, if the Hawkes model process is correct, the transformed inter-arrival times on dimension ℓ, $z_i^\ell = 1 - \exp\left\{-\left[\Lambda_\ell\left(t_i^\ell\right) - \Lambda_\ell\left(t_{i-1}^\ell\right)\right]\right\}$, where $\Lambda_\ell(t) = \int_0^t \lambda_\ell(s)ds$ is the compensator for $\lambda_\ell(t)$), follow a uniform distribution on the unit interval. Quantile-quantile plots of the (transformed) observed inter-arrival vs. those for the uniform distribution for all seven methods are shown in Figures 3.6 to 3.12. All seven methods produce very similar results. In all cases, the plots suggests that a MHP model tends to predict somewhat shorter inter-arrival times than expected.

The construction of the quantile-quantile plots for goodness-of-fit analysis of the real-world example relies on the well-known time-rescaling theorem [33]: if the Hawkes model process is correct, the transformed inter-arrival times on dimension ℓ, z_i^ℓ defined as follows:

$$1 - \exp\left\{-\left[\Lambda_\ell\left(t_i^\ell\right) - \Lambda_\ell\left(t_{i-1}^\ell\right)\right]\right\},$$

where $\Lambda_\ell(t) = \int_0^t \lambda_\ell(s)ds$ is the compensator for $\lambda_\ell(t)$), follow a uniform distribution on the unit interval. Quantile-quantile plots of the (transformed) observed inter-arrival vs. those for the uniform distribution for all seven methods are shown in Figures 3.6-3.12.

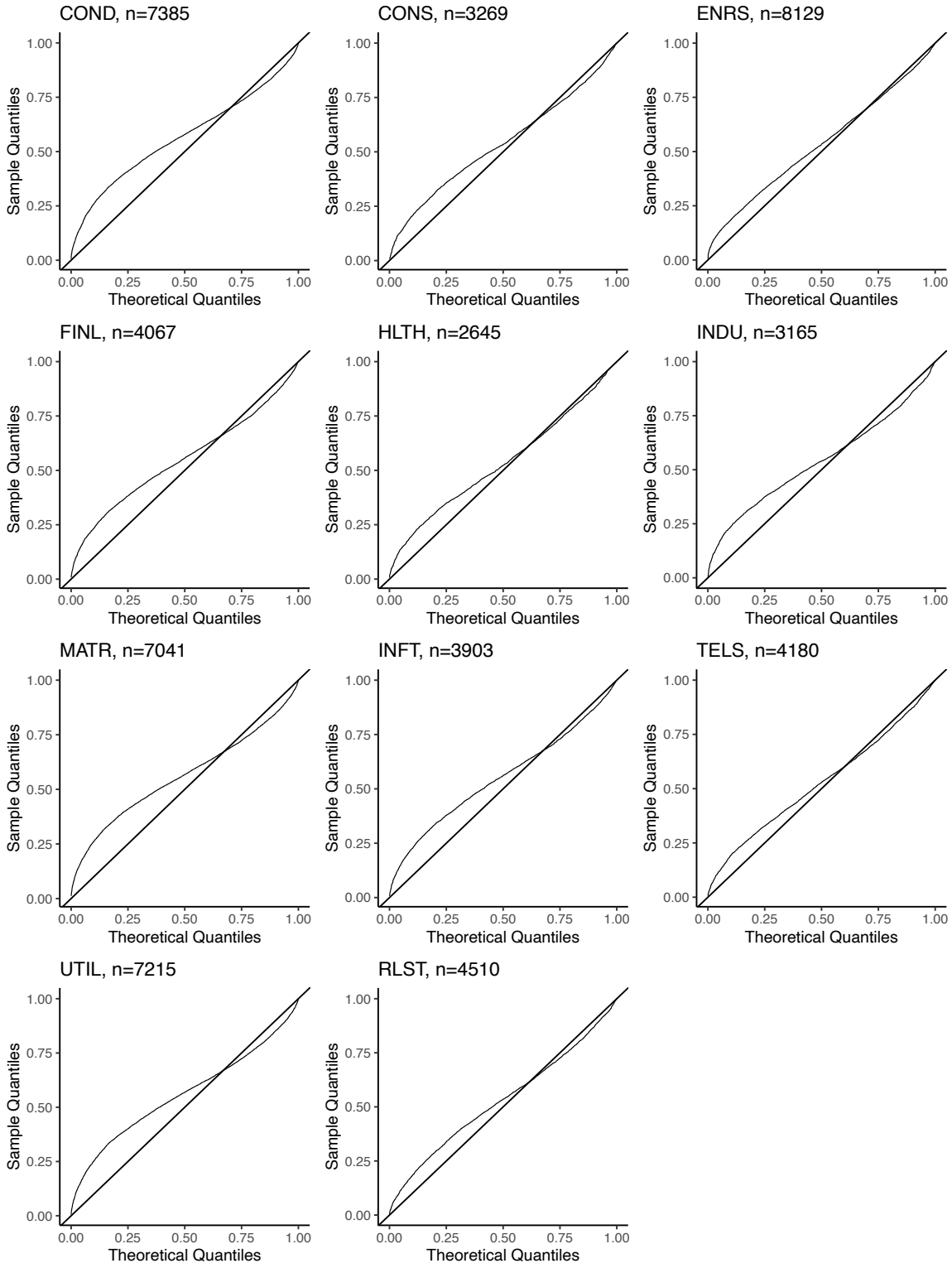

Figure 3.6: Quantile-quantile plots of the (transformed) observed inter-arrival vs. those for the uniform distribution for the SGEM method.

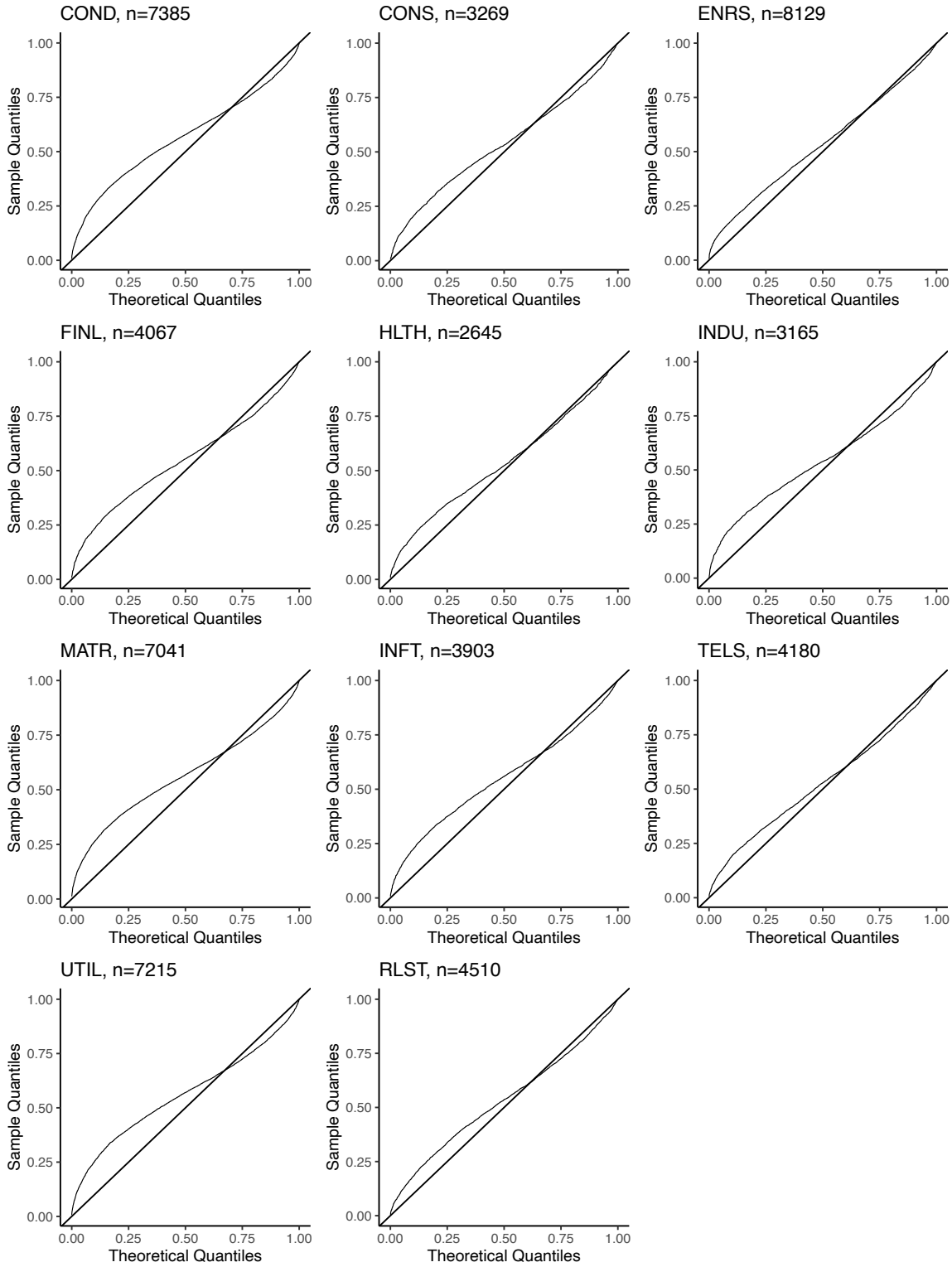

Figure 3.7: Quantile-quantile plots of the (transformed) observed inter-arrival vs. those for the uniform distribution for the SGEM-c method.

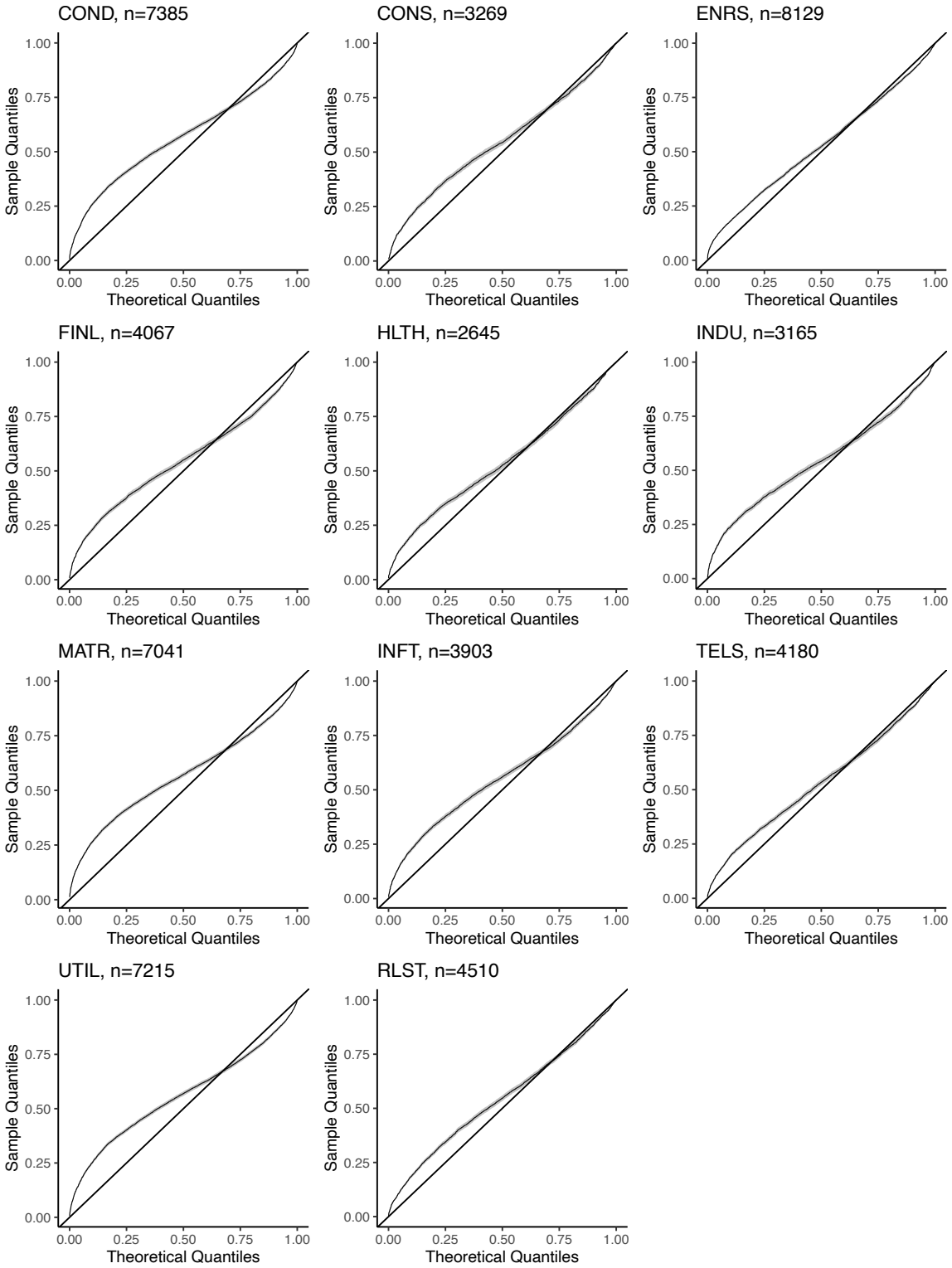

Figure 3.8: Quantile-quantile plots (along with 95% credible intervals) of the (transformed) observed inter-arrival vs. those for the uniform distribution for the SGVI method.

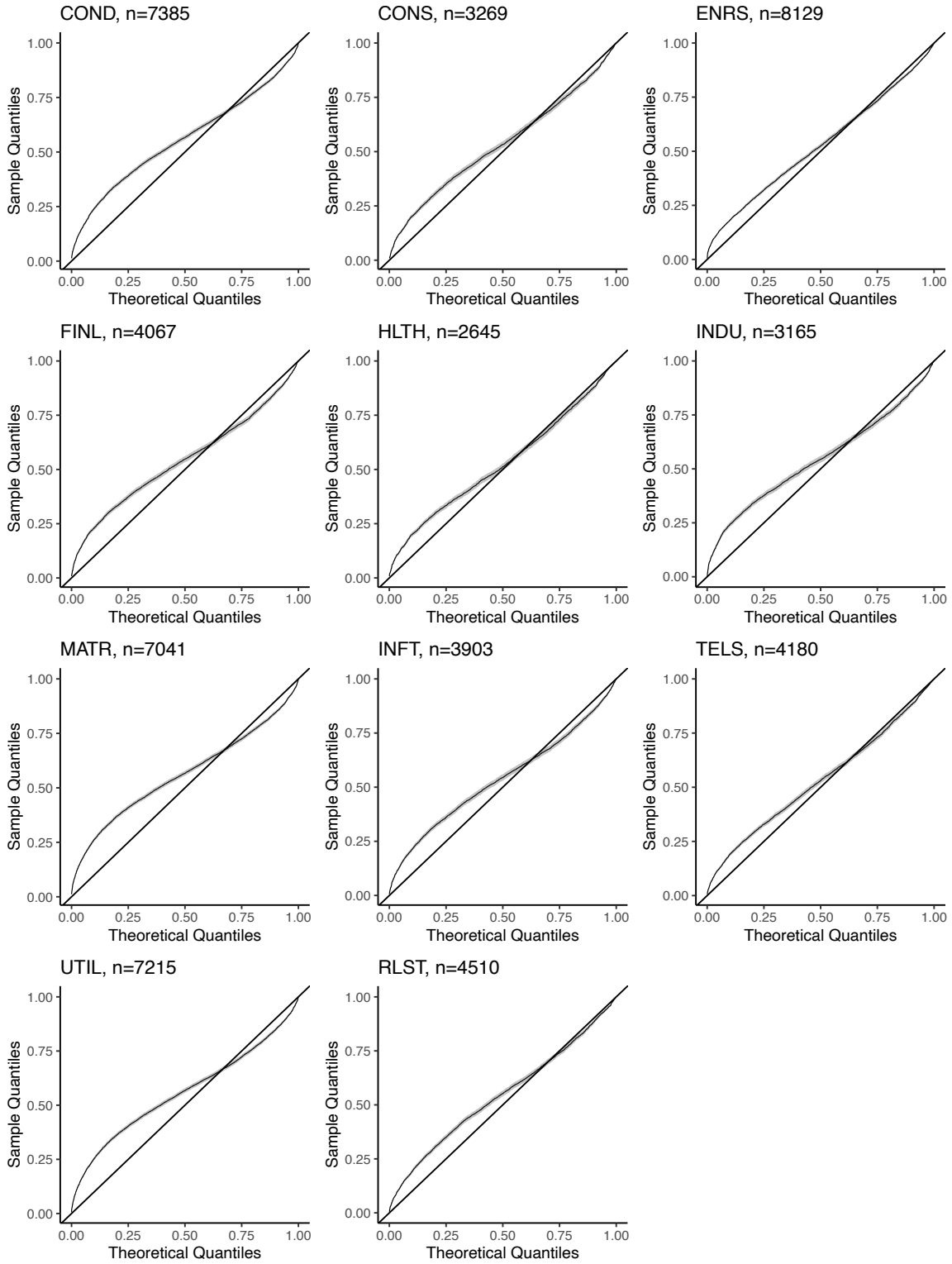

Figure 3.9: Quantile-quantile plots (along with 95% credible intervals) of the (transformed) observed inter-arrival vs. those for the uniform distribution for the SGVI-c method.

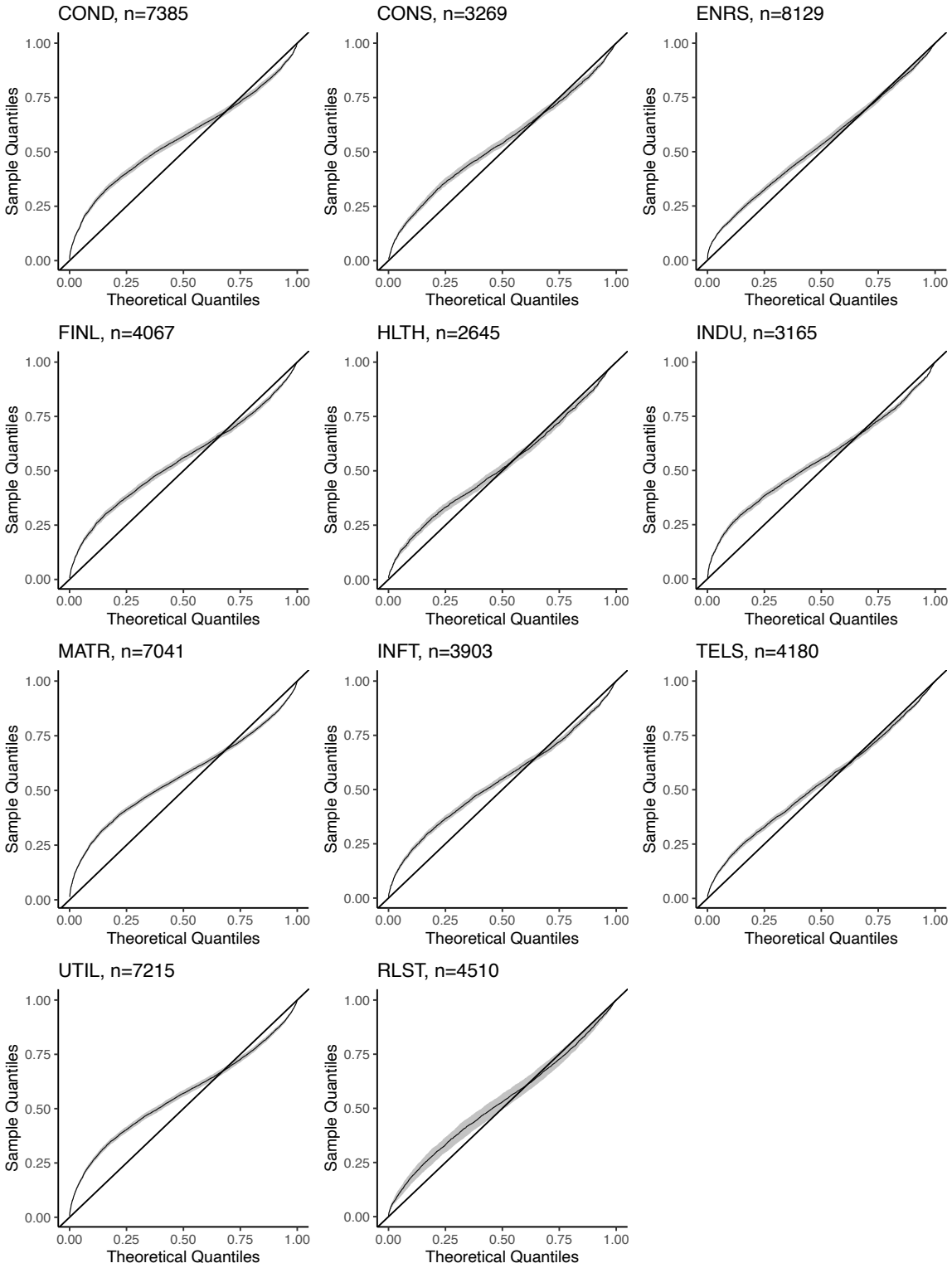

Figure 3.10: Quantile-quantile plots (along with 95% credible intervals) of the (transformed) observed inter-arrival vs. those for the uniform distribution for the SGLD method.

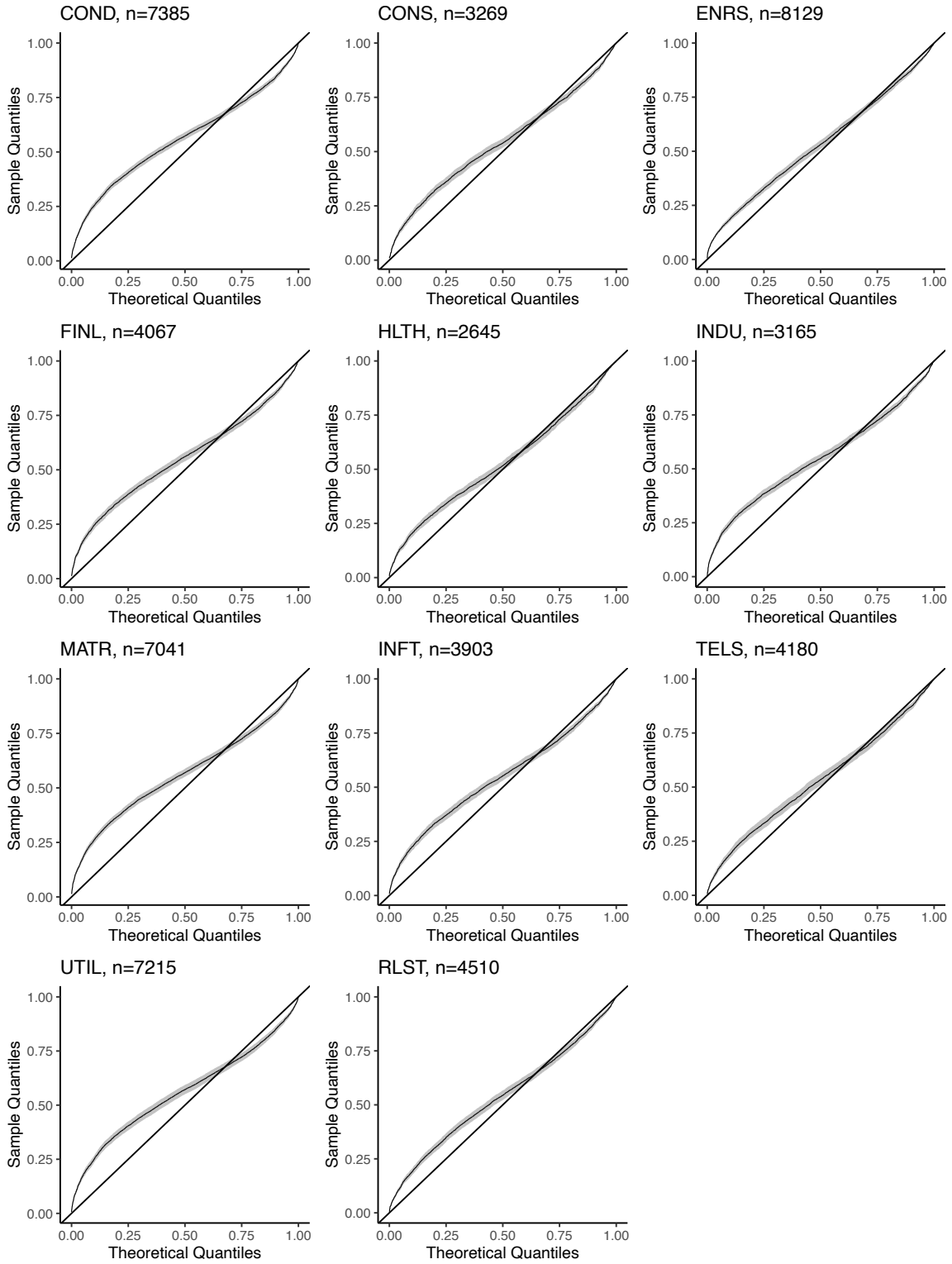

Figure 3.11: Quantile-quantile plots (along with 95% credible intervals) of the (transformed) observed inter-arrival vs. those for the uniform distribution for the MCMC method.

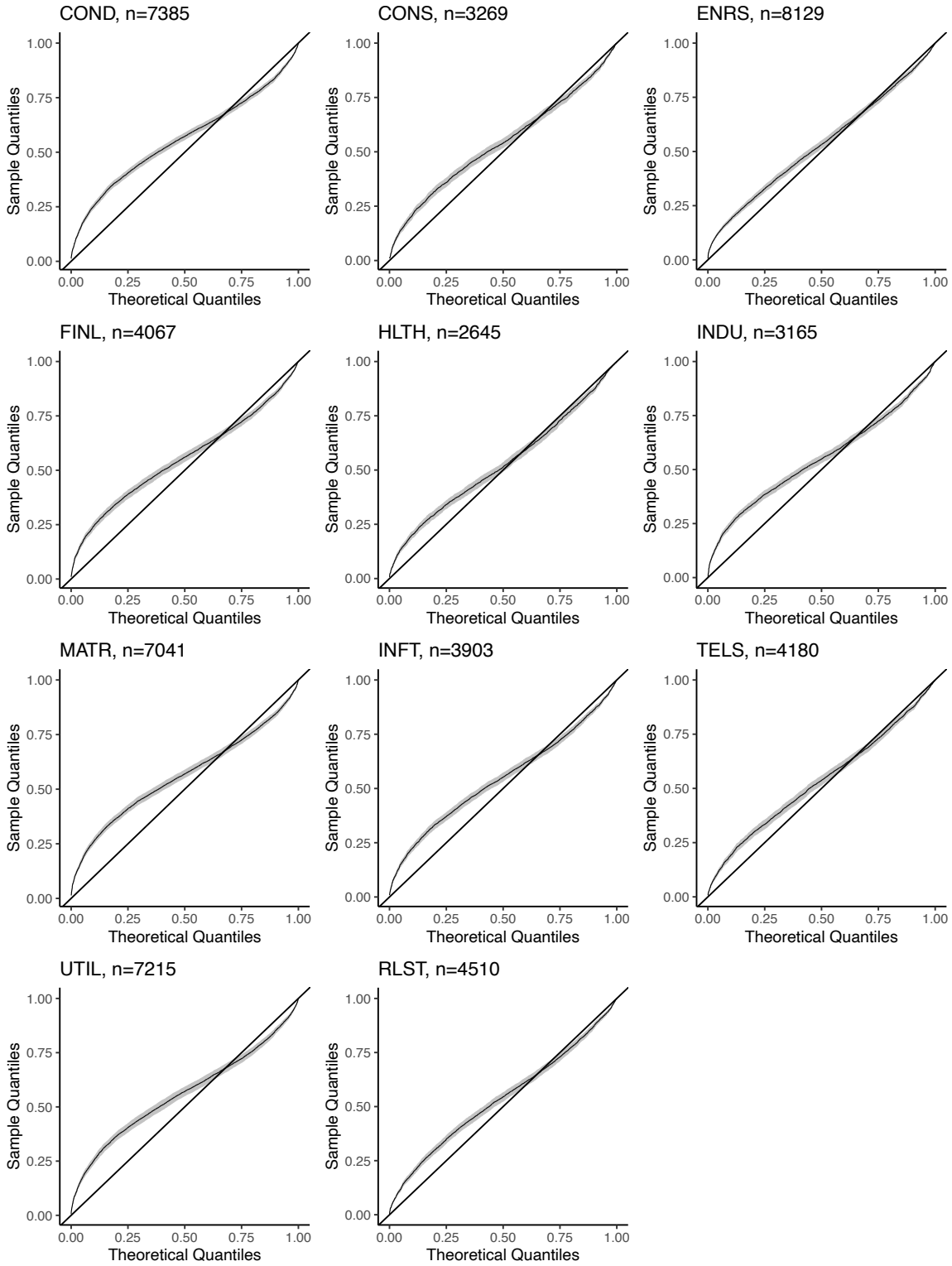

Figure 3.12: Quantile-quantile plots (along with 95% credible intervals) of the (transformed) observed inter-arrival vs. those for the uniform distribution for the MCMC-c method.

3.6 Discussion

Our experiments indicate that computational methods for Hawkes process models based on stochastic gradients can lead to accurate estimators and substantially reduce the computational burden in large datasets. However, our results also indicate some clear tradeoffs. SGEM algorithms are the fastest (which is consistent with the results of [169] for full-batch methods), but they do not yield interval estimates of the parameters. SGVI algorithms are almost as computationally efficient as SGEM and yield interval estimates. However, these interval estimates are too narrow, leading to substantial undercoverage. That variational inference underestimates the variance of the posterior distribution is well known (e.g., see 17), but it was still striking to see how low the coverage can be in the case of MHPs. SGLD algorithms are the slowest and require careful tuning, but can also lead to more accurate interval estimates if allowed to run for enough time.

Our experiments also suggest that the new approximation to the full-data likelihood based on a first-order Taylor expansion of the compensator of the Hawkes process has the potential to improve the accuracy of the algorithm with minimal additional computational costs. This was clearer for SGVI algorithms, where the approximation clearly improved the MSE of the point estimators. Finally, our experiments suggest that, as sample sizes grow, the fraction of time involved in the subsamples used to compute stochastic gradients can decrease as long as the number of observations in each subsample remains above a critical threshold. This is important, because it suggests that, at least empirically, the computational complexity of this algorithms can remain roughly constant as a function of the sample size.

An often important question in the context of MHPs is to identify which interactions are non-null. Bayesian inference for the the interaction graph can be accommodated through a slight modification of the priors in Section 3.2.2 in which the single Gamma prior for $\alpha_{k,\ell}$ is replaced by a "spike-and-slab" style specification (e.g., see 112) which, in this case, can be defined through a mixture of two Gamma priors with very different means. Extending our algorithms to this setting is straightforward, perhaps requiring the introduction of additional latent component indicators for each of the $\alpha_{k,\ell}$s. Our numerical explorations of

this variant of the model suggest that the SGEM and SGVI algorithms perform quite well in this setting, but the SGLD algorithm struggles to properly explore the parameter space. Another important question is how these methods extend to non-linear Hawkes processes (e.g., see [22]), which allows for the $\alpha_{k,\ell}$s to take negative values. With the addition of a nonlinear activation layer in the intensity function, nonlinear MHP models break model conjugacy and therefore cause computational challenges for SGEM and SGVI. For certain non-linear MHP models, latent variable augmentation can be used to restore model conjugacy, enabling a more or less direct extension of the algorithms described here (e.g., see [101, 168]). Alternatives that could be used in more general settings include methods based on Laplace approximations (e.g., see [161]) and non-conjuage message passing (e.g., see [84]), which have been successfully applied in other model settings. However, as far we are aware, these have not yet been explored for general non-linear Hawkes process. On the other hand, since SGLD does not rely on conjugacy, its extension to nonlinear MHP is relatively more straightforward than SGEM and SGVI.

This chapter focused on a very particular class of MHP with constant baseline intensity and a parametric excitation function. This was a deliberate choice meant to simplify exposition and interpretation. However, the insights from this manuscript apply much more broadly. For example, we are currently work on fast inference algorithms for MHP models where the excitation functions are modeled non parametrically using mixtures of dependent Dirichlet processes. This, and other extensions, will be discussed elsewhere.

methods	running time	0.05	0.1	0.2	0.3
SGEM	1 min	**0.999** (0.002)	0.998 (0.002)	0.994 (0.002)	0.993 (0.002)
	3 min	**1.001** (0.003)	0.999 (0.003)	0.997 (0.003)	0.996 (0.003)
	5 min	**1.001** (0.009)	1.000 (0.008)	0.999 (0.008)	0.997 (0.009)
SGEM-c	1 min	**1.003** (0.010)	0.998 (0.017)	0.996 (0.022)	1.000 (0.025)
	3 min	**1.006** (0.006)	1.003 (0.010)	0.994 (0.018)	0.994 (0.021)
	5 min	**1.007** (0.007)	1.006 (0.009)	1.003 (0.015)	1.002 (0.020)
SGVI	1 min	**0.999** (< 0.001)	0.996 (0.001)	0.991 (0.001)	0.988 (0.001)
	3 min	**1.001** (< 0.001)	0.999 (< 0.001)	0.995 (0.001)	0.992 (0.001)
	5 min	**1.001** (< 0.001)	1.000 (< 0.001)	0.997 (< 0.001)	0.994 (0.001)
SGVI-c	1 min	**0.998** (< 0.001)	0.995 (0.001)	0.991 (0.001)	0.988 (0.001)
	3 min	**1.000** (< 0.001)	0.998 (< 0.001)	0.994 (0.001)	0.992 (0.001)
	5 min	**1.000** (< 0.001)	0.999 (< 0.001)	0.996 (<0.001)	0.993 (0.001)
SGLD	1 min	**0.997** (0.001)	0.991 (0.002)	0.980 (0.003)	0.960 (0.005)
	3 min	**0.999** (0.001)	0.993 (0.001)	0.985 (0.002)	0.973 (0.003)
	5 min	**0.999** (0.001)	0.993 (0.001)	0.985 (0.002)	0.973 (0.003)

Table 3.6: Sensitivity analysis for the dataset sizes, with a large dataset ($T = 2000$) for the first data generation mechanism ($K = 3$). RBODLs for SGEM, SGVI and SGLD under running times of 1, 3 and 5 minutes, with $\kappa = 0.01$ being the reference subsampling ratio. Average RBODL across 50 datasets is shown, with standard deviations in the brackets.

methods	running time	0.05	0.1	0.2	0.3
SGEM	1 min	**1.008** (0.008)	1.008 (0.005)	1.008 (0.005)	1.007 (0.005)
	3 min	1.007 (0.007)	**1.008** (0.006)	1.008 (0.007)	1.007 (0.006)
	5 min	1.008 (0.01)	1.008 (0.009)	**1.009** (0.009)	1.008 (0.009)
SGEM-c	1 min	**1.003** (0.012)	1.003 (0.012)	1.002 (0.011)	1.001 (0.011)
	3 min	**1.003** (0.007)	1.003 (0.007)	1.004 (0.007)	1.003 (0.007)
	5 min	1.002 (0.006)	**1.003** (0.006)	1.003 (0.005)	1.002 (0.006)
SGVI	1 min	**1.032** (0.009)	1.032 (0.009)	1.03 (0.009)	1.029 (0.009)
	3 min	**1.032** (0.009)	1.032 (0.009)	1.032 (0.009)	1.031 (0.009)
	5 min	**1.033** (0.009)	1.033 (0.009)	1.033 (0.009)	1.032 (0.009)
SGVI-c	1 min	**1.021** (0.007)	1.021 (0.007)	1.019 (0.007)	1.018 (0.007)
	3 min	**1.022** (0.007)	1.022 (0.007)	1.022 (0.007)	1.021 (0.007)
	5 min	**1.022** (0.007)	1.022 (0.007)	1.022 (0.007)	1.022 (0.007)
SGLD	1 min	**1.017** (0.006)	1.015 (0.006)	1.008 (0.005)	1.004 (0.005)
	3 min	**1.019** (0.006)	1.018 (0.006)	1.014 (0.006)	1.009 (0.005)
	5 min	**1.020** (0.006)	1.020 (0.006)	1.017 (0.006)	1.011 (0.005)

Table 3.7: Sensitivity analysis for the dataset sizes, with a large dataset ($T = 500$) for the first data generation mechanism ($K = 3$). RBODLs for SGEM, SGVI and SGLD under running times of 1, 3 and 5 minutes, with $\kappa = 0.01$ being the reference subsampling ratio. Average RBODL across 50 datasets is shown, with standard deviations in the brackets.

methods	running time	0.05	0.1	0.2	0.3
SGEM	1 min	**1.003** (0.007)	1.002 (0.005)	1.000 (0.005)	0.997 (0.005)
	3 min	**1.004** (0.006)	1.003 (0.005)	1.002 (0.006)	1.000 (0.006)
	5 min	1.003 (0.008)	**1.004** (0.008)	1.003 (0.008)	1.001 (0.008)
SGEM-c	1 min	**1.003** (0.005)	1.003 (0.005)	1.000 (0.004)	0.998 (0.004)
	3 min	1.003 (0.009)	**1.004** (0.008)	1.003 (0.008)	1.001 (0.008)
	5 min	**1.004** (0.006)	1.003 (0.006)	1.003 (0.006)	1.002 (0.006)
SGVI	1 min	**1.003** (0.005)	1.001 (0.005)	0.998 (0.005)	0.996 (0.005)
	3 min	1.003 (0.006)	**1.004** (0.006)	1.002 (0.006)	1.000 (0.007)
	5 min	**1.004** (0.006)	1.004 (0.006)	1.003 (0.006)	1.001 (0.006)
SGVI-c	1 min	**1.003** (0.003)	1.003 (0.002)	1.000 (0.002)	0.998 (0.002)
	3 min	**1.004** (0.006)	1.004 (0.005)	1.003 (0.005)	1.002 (0.005)
	5 min	**1.003** (0.012)	1.003 (0.012)	1.003 (0.012)	1.002 (0.012)
SGLD	1 min	**1.004** (0.001)	1.001 (0.001)	0.996 (0.001)	0.993 (0.001)
	3 min	**1.003** (0.001)	1.003 (0.001)	1.000 (0.001)	0.997 (0.001)
	5 min	**1.003** (0.001)	1.003 (0.001)	1.001 (0.001)	0.999 (0.001)

Table 3.8: Sensitivity analysis for the stochastic search parameters for the first data generation mechanism ($K = 3$), with $\tau_1 = 5, \tau_2 = 0.51$. RBODLs for SGEM, SGVI and SGLD under running times of 1, 3 and 5 minutes, with $\kappa = 0.01$ being the reference subsampling ratio. Average RBODL across 50 datasets is shown, with standard deviations in the brackets.

methods	running time	0.05	0.1	0.2	0.3
SGEM	1 min	**1.003** (0.007)	1.002 (0.005)	1.000 (0.005)	0.997 (0.005)
	3 min	**1.004** (0.006)	1.003 (0.005)	1.002 (0.006)	1.000 (0.006)
	5 min	1.003 (0.008)	**1.004** (0.008)	1.003 (0.008)	1.001 (0.008)
SGEM-c	1 min	**1.003** (0.005)	1.003 (0.005)	1.000 (0.004)	0.998 (0.004)
	3 min	1.003 (0.009)	**1.004** (0.008)	1.003 (0.008)	1.001 (0.008)
	5 min	**1.004** (0.006)	1.003 (0.006)	1.003 (0.006)	1.002 (0.006)
SGVI	1 min	**1.003** (0.005)	1.001 (0.005)	0.998 (0.005)	0.996 (0.005)
	3 min	1.003 (0.006)	**1.004** (0.006)	1.002 (0.006)	1.000 (0.007)
	5 min	1.004 (0.006)	**1.004** (0.006)	1.003 (0.006)	1.001 (0.006)
SGVI-c	1 min	**1.003** (0.003)	1.003 (0.002)	1.000 (0.002)	0.998 (0.002)
	3 min	**1.004** (0.006)	1.004 (0.005)	1.003 (0.005)	1.002 (0.005)
	5 min	**1.003** (0.012)	1.003 (0.012)	1.003 (0.012)	1.002 (0.012)
SGLD	1 min	**1.004** (0.001)	1.001 (0.001)	0.996 (0.001)	0.993 (0.001)
	3 min	**1.003** (0.001)	1.003 (0.001)	1.000 (0.001)	0.997 (0.001)
	5 min	**1.003** (0.001)	1.003 (0.001)	1.001 (0.001)	0.999 (0.001)

Table 3.9: Sensitivity analysis for the stochastic search parameters for the first data generation mechanism ($K = 3$), with $\tau_1 = 1, \tau_2 = 1$. RBODLs for SGEM, SGVI and SGLD under running times of 1, 3 and 5 minutes, with $\kappa = 0.01$ being the reference subsampling ratio. Average RBODL across 50 datasets is shown, with standard deviations in the brackets.

methods	running time	0.05	0.1	0.2	0.3
SGEM	1 min	**1.000** (0.006)	0.999 (0.006)	0.997 (0.007)	0.997 (0.006)
	3 min	0.999 (0.009)	**1.000** (0.009)	0.997 (0.009)	0.997 (0.009)
	5 min	**1.000** (0.007)	0.998 (0.006)	0.998 (0.007)	0.997 (0.007)
SGEM-c	1 min	**1.000** (0.009)	1.000 (0.010)	1.000 (0.01)	0.999 (0.011)
	3 min	**1.000** (0.005)	1.000 (0.004)	0.999 (0.005)	0.999 (0.005)
	5 min	**1.001** (0.009)	1.000 (0.009)	1.000 (0.009)	0.999 (0.009)
SGVI	1 min	**0.998** (0.002)	0.995 (0.003)	0.991 (0.004)	0.989 (0.005)
	3 min	**0.998** (0.002)	0.996 (0.003)	0.993 (0.004)	0.992 (0.004)
	5 min	**0.999** (0.002)	0.997 (0.003)	0.995 (0.003)	0.993 (0.004)
SGVI-c	1 min	**0.996** (0.002)	0.993 (0.003)	0.990 (0.004)	0.988 (0.005)
	3 min	**0.997** (0.002)	0.994 (0.003)	0.992 (0.004)	0.990 (0.004)
	5 min	**0.997** (0.002)	0.995 (0.003)	0.992 (0.003)	0.991 (0.004)
SGLD	1 min	**1.001** (0.001)	1.000 (0.004)	0.997 (0.010)	0.989 (0.017)
	3 min	**1.001** (0.001)	1.000 (0.003)	0.998 (0.008)	0.995 (0.013)
	5 min	**1.001** (0.001)	1.000 (0.003)	0.998 (0.006)	0.995 (0.011)

Table 3.10: Sensitivity analysis for the stochastic search parameters for the first data generation mechanism ($K = 3$), with $\tau_1 = 1, \tau_2 = 5$. RBODLs for SGEM, SGVI and SGLD under running times of 1, 3 and 5 minutes, with $\kappa = 0.01$ being the reference subsampling ratio. Average RBODL across 50 datasets is shown, with standard deviations in the brackets.

Methods	RMISE $(\boldsymbol{\alpha}, \boldsymbol{\beta})$	MAE $(\boldsymbol{\mu})$	IS	ACR	AIW
SGLD	0.052	0.109	3.898	0.667	0.844
	(0.019)	(0.283)	(4.739)	(0.152)	(0.172)
SGVI	0.046	0.103	6.163	0.333	0.222
	(0.008)	(0.048)	(2.117)	(0.131)	(0.012)
SGVI-c	0.040	0.093	4.905	0.429	0.213
	(0.007)	(0.044)	(1.698)	(0.139)	(0.012)
SGEM	0.049	0.044	-	-	-
	(0.008)	(0.042)			
SGEM-c	0.046	0.038	-	-	-
	(0.008)	(0.037)			

Table 3.11: Estimation metrics across all seven methods under stochastic search parameters ($\tau_1 = 1, \tau_2 = 0.51$) for the first data generation mechanism ($K = 3$). The values in the grid cells are the average across 50 datasets, with the standard deviation in the brackets.

Methods	RMISE $(\boldsymbol{\alpha}, \boldsymbol{\beta})$	MAE $(\boldsymbol{\mu})$	IS	ACR	AIW
SGLD	0.053	0.108	3.912	0.667	0.907
	(0.014)	(0.046)	(3.118)	(0.129)	(0.193)
SGVI	0.046	0.105	6.359	0.333	0.222
	(0.008)	(0.048)	(2.104)	(0.138)	(0.012)
SGVI-c	0.040	0.095	4.797	0.429	0.216
	(0.007)	(0.044)	(1.695)	(0.139)	(0.012)
SGEM	0.050	0.046	-	-	-
	(0.008)	(0.040)			
SGEM-c	0.049	0.044	-	-	-
	(0.008)	(0.040)			

Table 3.12: Estimation metrics across all seven methods under stochastic search parameters ($\tau_1 = 5, \tau_2 = 0.51$) for the first data generation mechanism ($K = 3$). The values in the grid cells are the average across 50 datasets, with the standard deviation in the brackets.

$(\tau_1 = 1, \tau_2 = 1)$					
methods	RMISE $(\boldsymbol{\alpha}, \boldsymbol{\beta})$	RMSE($\boldsymbol{\mu}$)	IS	ACR	AIW
SGLD	0.137	0.264	33.020	0.095	0.231
	(0.033)	(0.528)	(70.245)	(0.055)	(0.218)
SGVI	0.158	0.182	41.842	0.095	0.291
	(0.024)	(0.061)	(12.961)	(0.046)	(0.087)
SGVI-c	0.152	0.176	36.011	0.095	0.259
	(0.026)	(0.061)	(11.889)	(0.055)	(0.070)
SGEM	0.087	0.136	-	-	-
	(0.061)	(0.062)			
SGEM-c	0.113	0.147	-	-	-
	(0.061)	(0.054)			

Table 3.13: Estimation metrics across all seven methods under different sets of stochastic search parameters ($\tau_1 = 1, \tau_2 = 1$) for the first data generation mechanism ($K = 3$). The values in the grid cells are the average across 50 datasets, with the standard deviation in the brackets.

$(\tau_1 = 5, \tau_2 = 1)$					
methods	RMISE $(\boldsymbol{\alpha}, \boldsymbol{\beta})$	RMSE($\boldsymbol{\mu}$)	IS	ACR	AIW
SGLD	0.150	0.332	37.133	0.048	0.250
	(0.033)	(0.121)	(66.530)	(0.056)	(0.230)
SGVI	0.158	0.182	41.879	0.095	0.291
	(0.025)	(0.061)	(13.020)	(0.046)	(0.088)
SGVI-c	0.152	0.176	36.011	0.095	0.259
	(0.026)	(0.061)	(11.889)	(0.055)	(0.070)
SGEM	0.113	0.140	-	-	-
	(0.061)	(0.150)			
SGEM-c	0.151	0.114	-	-	-
	(0.043)	(0.062)			

Table 3.14: Estimation metrics across all seven methods under different sets of stochastic search parameters $(\tau_1 = 5, \tau_2 = 1)$ for the first data generation mechanism $(K = 3)$. The values in the grid cells are the average across 50 datasets, with the standard deviation in the brackets.

		RMISE ($\boldsymbol{\alpha}, \boldsymbol{\beta}$)	MAE ($\boldsymbol{\mu}$)	IS	ACR	AIW
	$r = 0.5$	0.050 (0.008)	0.097 (0.048)	6.030 (1.420)	0.333 (0.078)	0.220 (0.009)
	$r = 1$	0.040 (0.007)	0.093 (0.044)	4.905 (1.698)	0.429 (0.139)	0.213 (0.012)
SGVI-c	$r = 2$	0.039 (0.007)	0.064 (0.038)	2.609 (1.217)	0.476 (0.097)	0.200 (0.011)
	$r = 3$	0.052 (0.007)	0.077 (0.037)	5.674 (0.770)	0.357 (0.108)	0.180 (0.012)
	$r = 4$	0.080 (0.006)	0.100 (0.045)	13.374 (1.996)	0.238 (0.123)	0.161 (0.012)
	$r = 0.5$	0.050 (0.008)	0.038 (0.040)	-	-	-
	$r = 1$	0.046 (0.008)	0.038 (0.037)	-	-	-
SGEM-c	$r = 2$	0.039 (0.007)	0.024 (0.029)	-	-	-
	$r = 3$	0.048 (0.007)	0.019 (0.025)	-	-	-
	$r = 4$	0.073 (0.007)	0.028 (0.034)	-	-	-

Table 3.15: Estimation metrics for SGVI-c and SGEM-c under different values of r for the first data generation mechanism ($K = 3$). The values in the grid cells are the average across 50 datasets, with the standard deviation in the brackets.

	MCMC-c	MCMC	SGVI-c	SGVI	SGEM-c	SEM	SGLD
MCMC-c	0.000	0.018	0.302	0.272	0.263	0.252	0.588
MCMC	0.018	0.000	0.294	0.246	0.237	0.228	0.565
SGVI-c	0.302	0.294	0.000	0.294	0.280	0.272	0.485
SGVI	0.272	0.246	0.294	0.000	0.017	0.013	0.565
SGEM-c	0.263	0.237	0.280	0.017	0.000	0.003	0.582
SEM	0.252	0.228	0.272	0.013	0.003	0.000	0.573
SGLD	0.588	0.565	0.485	0.565	0.582	0.573	0.000

Table 3.16: Mean square distance between the point estimates for $\boldsymbol{\alpha}$ among the seven methods, computed after log transformation.

	MCMC-c	MCMC	SGVI-c	SGVI	SGEM-c	SEM	SGLD
MCMC-c	0.000	0.486	2.385	2.568	2.540	2.596	3.541
MCMC	0.486	0.000	2.596	2.531	2.577	2.577	3.745
SGVI-c	2.385	2.596	0.000	2.373	2.264	2.375	4.290
SGVI	2.568	2.531	2.373	0.000	0.078	0.053	3.435
SGEM-c	2.540	2.577	2.264	0.078	0.000	0.021	3.267
SEM	2.596	2.577	2.375	0.053	0.021	0.000	3.462
SGLD	3.541	3.745	4.290	3.435	3.267	3.462	0.000

Table 3.17: Mean square distance between the point estimates for $\boldsymbol{\beta}$ among the seven methods, computed after log transformation.

	MCMC-c	MCMC	SGVI-c	SGVI	SGEM-c	SEM	SGLD
MCMC-c	0.000	0.007	0.090	0.097	0.142	0.129	0.219
MCMC	0.007	0.000	0.114	0.115	0.175	0.161	0.212
SGVI-c	0.090	0.114	0.000	0.061	0.083	0.072	0.270
SGVI	0.097	0.115	0.061	0.000	0.045	0.034	0.324
SGEM-c	0.142	0.175	0.083	0.045	0.000	0.001	0.466
SEM	0.129	0.161	0.072	0.034	0.001	0.000	0.435
SGLD	0.219	0.212	0.270	0.324	0.466	0.435	0.000

Table 3.18: Mean square distance between the point estimates for $\boldsymbol{\mu}$ among the seven methods, computed after log transformation.

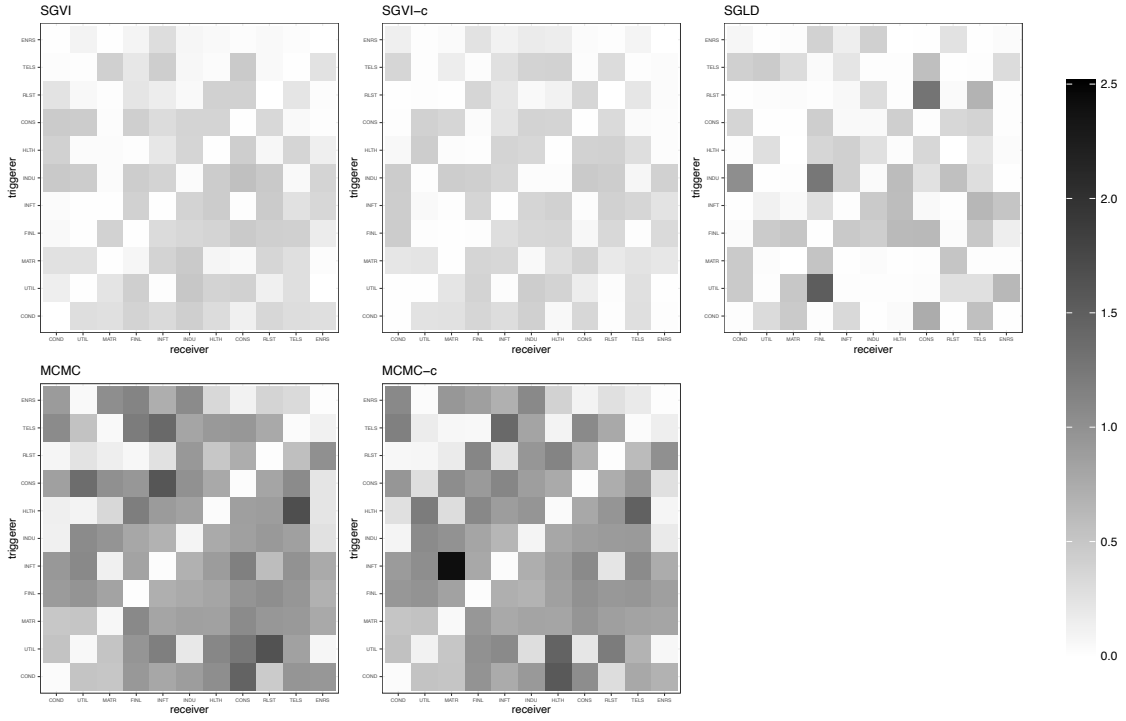

Figure 3.13: Heatmaps for 95% credible interval lengths estimates of $\boldsymbol{\beta}$ parameters for the corresponding 11 sectors, for all five algorithms that yield uncertainty estimates.

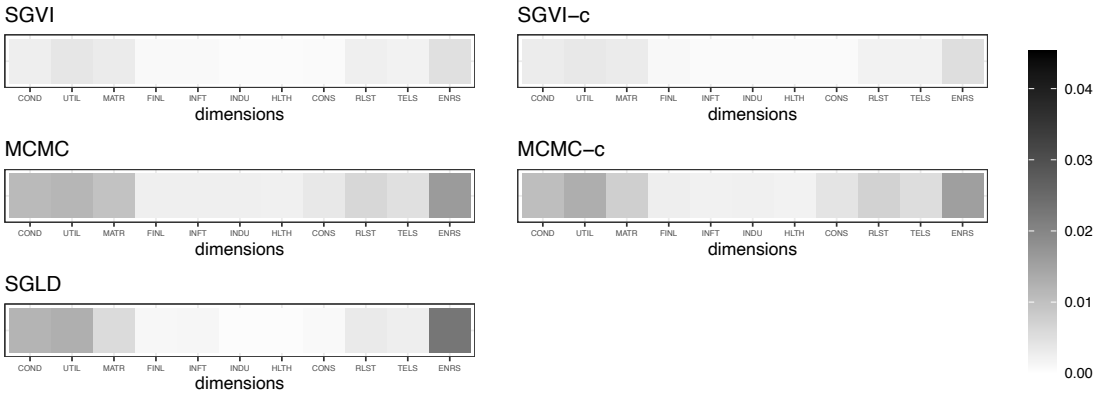

Figure 3.14: Heatmaps for 95% credible interval lengths estimates of $\boldsymbol{\mu}$ parameters for the corresponding 11 sectors, for all five algorithms that yield uncertainty estimates.

Chapter 4

SEMIPARAMETRIC ESTIMATION FOR MULTIVARIATE HAWKES PROCESSES USING DEPENDENT DIRICHLET PROCESSES: AN APPLICATION TO ORDER FLOW DATA IN FINANCIAL MARKETS

4.1 Introduction

In the past two decades, the proliferation of electronic trading systems has led to financial instruments such as stocks and futures being more frequently traded in order-driven markets. In these order-driven markets, buy and sell orders are submitted by traders to an electronic platform, which matches them with the best available offers (e.g., see [63]). Due to the nature of electronic trading, the arrival of orders tends to have high-frequency, with orders usually arriving within milliseconds of each other. The order flow in such markets is of particular research interest, as it not only provides insights to high-frequency trading and order execution strategies [3], but also aids in understanding the interplay of supply and demand and their roles in price formation [31].

A number of approaches have been proposed to analyze data from order-driven markets, including queueing systems [117], stochastic partial differential equations [32] and Hawkes processes [129]. In Chapter 4 we focus on multivariate Hawkes Processes (MHPs, e.g., see [67] and [95]) models. MHPs have been widely used to model sequences of events that demonstrate *self-* and *mutually-exciting* behaviors, i.e., patterns in which the likelihood of events increases after the occurrence of others. MHPs can be characterized through their conditional intensity functions, which describe the instantaneous rate of arrivals of new events. The excitation function is a key module of the conditional intensity function, as it controls how past events cause the conditional intensity function to change, and how such intensity decays when no new events are observed in the meantime. Excitation functions are often modeled parametrically (e.g., using exponential or polynomial functions) that assume monotonic excitation decay [67]. However, the inter-event times in many real-world examples tend to have complex patterns, motivating the need for flexible excitation functions

(e.g., see 102 and 140).

Nonparametric models for the intensity function associated with a Hawkes processes have been investigated in the past. For example, [103] proposed a histogram estimator that can be computed via the expectation-maximization (EM) algorithm. The efficacy of their method is further studied in [145]. In related work, [90] and [170] developed penalized maximum likelihood where the regularization term favors smooth estimates of the intensity functions. Alternatively, [133] developed penalized projection estimators. Their work was extended by [65], who provided non-asymptotic theoretical guarantees on model selection. On yet another track, [7] and [8] proposed a nonparametric estimation framework for MHP models based on the discretization of a Wiener-Hopf system that relates the excitation functions to the second order statistics of the MHP. Under a Bayesian framework, [166] and [169] considered flexible models for the excitation functions through Gaussian process priors coupled with squared or sigmoid link functions. These methods lead to flexible models for the excitation function, but their theoretical properties are not well understood. In contrast, [42] modeled the excitation functions using nonparametric mixtures and established posterior concentration rates for Bayesian nonparametric estimation of MHP, under the finite support assumption for the excitation functions. A related approach was developed in parallel in [102].

In this Chapter, we introduce a Bayesian semiparametric model for MHPs that builds on the ideas of [42] and [102] but addresses various practical questions that have so far remained open. One challenge associated with the estimation of MHPs is that the number of parameters grows quadratically with the number of dimensions. Hence, with a dataset of moderate size, modeling each dimension of the process independently can lead to inefficiencies. Motivated by the observation that excitation functions often look similar across different dimensions, we propose a hierarchical modeling approach based on mixtures of nonparametric mixtures. More specifically, we adapt the approach introduced in [115], which models each of the excitation functions as a mixture of an idiosyncratic component and a common component shared by all excitation functions, and study some of the properties of such formulation. A second challenge in implementing MHP models is computational in nature. As is the case more generally, Markov chain Monte Carlo (MCMC) algorithms

are the most common approach to computation for Bayesian models for Hawkes processes (e.g., see [131]). However, MCMC algorithms for Hawkes process models are often too slow even for moderate sample sizes because, except for special cases such as the exponential excitation function, their complexity is quadratic in both the number of observations and the number of dimensions. This challenge is amplified in the case of nonparametric models. To address it, we expand on previous work on the use of stochastic gradient methods for MHPs (e.g., see [79]), and develop a scalable stochastic variational inference (SVI) algorithm that can be used to fit our model. An important part of this development involves a carefully comparison of the performance of SVI and MCMC methods, both in terms of accuracy and speed. Furthermore, to illustrate the performance of the model, we applied our method to study the limit order book data for Amazon obtained from LOBSTER [73]. Our analysis indicates that some of the excitation functions that arise from the analysis of this kind of data can have features such as non-monotonicity that are not captured by the kind of parametric forms that are commonly used in practice.

In summary, we make three key contributions in this chapter: (1) we propose a novel and flexible model for linear MHPs in which the various excitation functions are assigned a joint nonparametric prior that allows us to efficiently borrow information, (2) we develop MCMC and SVI algorithms for estimation and prediction in the context of this model, and thoroughly evaluate their relative performance, and (3) we illustrate the need for both nonparametric inference and fast computation in the context of a real-world application.

The remainder of Chapter 4 is structured as follows. In Section 2 provides a brief review of multivariate Hawkes process. Section 3 outlines our model and discusses some of its properties. Section 4 describes the two computing algorithms we employ for our model, a Markov chain Monte Carlo algorithm and a stochastic gradient variational approximation. Sections 5 and 6 discuss the results from simulations and real-world applications. Finally, Section 7 presents our conclusions and points out potential future directions for research.

4.2 Multivariate Hawkes Processes

Let $N^{(1)}(t), \ldots, N^{(K)}(t)$ be a collection of K point processes defined on the positive real line $\mathbb{R}^+$, where $N^{(k)}(t)$ represents the number of events on dimension k that occur on the

interval $[0, t]$. We denote a generic set of observations from this process by $\mathbf{X} = \{(t_i, d_i) : i = 1, \ldots, n\}$, where $t_i \in \mathcal{R}^+$ represents the timestamp at which the i-th event occurs, and $d_i \in \{1, \ldots, K\}$ represents the dimension in which it occurs. Then, $\mathbf{X}$ follows a multivariate Hawkes process [67, 95] if the conditional intensity function on dimension k has the following form:

$$\lambda_k(t) \equiv \lim_{h \to 0} \frac{\mathbb{E}\left[N^{(k)}(t+h) - N^{(k)}(t) \mid \mathcal{M}_t\right]}{h} = \mu_k + \sum_{\ell=1}^{K} \sum_{\{i:t_i<t, d_i=k\}} \alpha_{\ell,k} \tilde{\phi}_{\ell,k}\left(t - t_i\right),$$

where $\mathcal{M}_t$ denotes the subset of $\mathbf{X}$ for which $t_i < t$, $\mu_k > 0$ is the background intensity for dimension k, $\alpha_{\ell,k} > 0$ is the parameter that controls the strength by which past events from dimension ℓ influence the occurrence of new events on dimension k, and $\tilde{\phi}_{\ell,k}(\cdot) : \mathbf{R}^+ \to \mathbf{R}^+$ is the (normalized) excitation function that controls how such influence decays over time. Note that we require that $\int_0^\infty \tilde{\phi}_{\ell,k,}(s)\mathrm{d}s = 1$, which ensures that $\tilde{\phi}_{\ell,k}$ and $\alpha_{\ell,k}$ are identifiable. The log-likelihood for the MHP can then be expressed as [33]:

$$\begin{aligned}
\mathcal{L}\left(\mathbf{X} \mid \{\mu_k\}, \{\alpha_{k,\ell}\}, \{\phi_{k,\ell}\}\right) &= \sum_{k=1}^{K} \sum_{d_i=k} \log \lambda_k\left(t_i\right) - \sum_{k=1}^{K} \int_0^T \lambda_k(s) ds \\
&= \sum_{k=1}^{K} \sum_{d_i=k} \log \left(\mu_k + \sum_{k=1}^{K} \sum_{\substack{\{(i,j:j<i \\ d_j=k, d_i=\ell\}}} \alpha_{\ell,k} \tilde{\phi}_{\ell,k}(t_i - t_j) \right) - \sum_{k=1}^{K} \mu_k T \\
&\qquad - \sum_{k=1}^{K} \sum_{\ell=1}^{K} \alpha_{\ell,k} \sum_{\{i:d_i=k\}} \tilde{\Phi}_{\ell,k}(T - t_i),
\end{aligned} \tag{4.1}$$

where $\tilde{\Phi}_{\ell,k}(t) = \int_0^t \tilde{\phi}_{\ell,k}(s) ds$.

An alternative construction of the MHP is as a multivariate branching process in which the first generation of events in dimension k (often called "immigrants" in the literature) arise from a homogeneous Poisson process with rate μ_k, and the points in subsequent generations are generated from non-homogenous Poisson processes with rates given by the $\alpha_{\ell,k}$s and the interarrival times are controlled by the $\phi_{\ell,k}$s. See [68] and [95] for details. While the branching structure is typically latent (i.e., not observed), this construction both provides insight into the properties of the model and can be exploited to facilitate computation. For example, this construction makes it clear that an MHP is stationary if and only if the

spectral radius of the matrix $[\alpha_{\ell,k}]$ is less than 1. In the sequel, we use the binary matrix $\boldsymbol{B}$ where

$$B_{j,j} = \begin{cases} 1 & \text{the } j\text{-th event is an immigrant} \\ 0 & \text{otherwise,} \end{cases}$$

$$B_{i,j} = \begin{cases} 1 & \text{the } j\text{-th event is an offspring of the } i\text{-th event} \\ 0 & \text{otherwise,} \end{cases}$$

to encode the latent branching structure associated with a realization of an MHP. The augmented likelihood for the data is then given by

$$\mathcal{L}\left(\mathbf{X}, \boldsymbol{B} \mid \{\mu_k\}, \{\alpha_{k,\ell}\}, \{\phi_{k,\ell}\}\right) = \sum_{k=1}^{K}\sum_{\ell=1}^{K}\left[|O_{k,\ell}|\left(\log \alpha_{k,\ell}\right) - \sum_{\substack{\{(i,j):i<j \\ d_i=k, d_j=\ell\}}} B_{i,j}\tilde{\phi}_{k,\ell}(t_j - t_i)\right] + \sum_{\ell=1}^{K}|I_\ell|\log\mu_\ell - \sum_{\ell=1}^{K}\mu_\ell T - \sum_{k=1}^{K}\sum_{\ell=1}^{K}\alpha_{k,\ell}\sum_{\{i:d_i=k\}}\tilde{\Phi}_{k,\ell}(T-t_i), \quad (4.2)$$

where $|I_\ell| = \sum_{d_i=\ell} I(B_{ii}=1)$ and $|O_{k,\ell}| = \sum_{d_i=k,d_j=\ell,i<j} I(B_{ij}=1)$ denote the number of immigrants for dimension k and the number of offpring on dimension k who arise from points on dimension ℓ.

4.3 Nonparametric Bayesian modeling of excitation functions for multivariate Hawkes processes

As we discussed in the introduction, efficient and flexible estimation of the excitation functions is one of the key challenges when working with MHPs. Because the normalized excitation functions are constrained to integrate out to one, this problem can be reduced to one of simultaneous estimation for a (finite) collection of densities.

The literature on Bayesian estimation for collections of densities is dominated by methods based on (dependent) nonparametric mixtures (e.g., see [139] and [128]). In the case of countable, exchangeable collections of densities, examples of such dependent nonparametric mixture models include the Hierarchical Dirichlet process (HDP, [152], the Nested Dirichlet process (NDP, [138]), mixture of mixture approaches such as those in [115], and the probit stick-breaking process [137], among others.

In the sequel, we adapt the methodology introduced in [115] to the estimation of excitation functions in multivariate Hawkes processes. More specifically, we consider mixtures of (scaled) Beta kernels of the form

$$\tilde{\phi}_{k,\ell}(t) = \int f_{\text{Beta}}(t \mid a, b, T_0) \mathrm{d}G_{k,\ell}(a, b), \tag{4.3}$$

where $\{G_{k,\ell}\}$ is a collection of almost-surely discrete mixing measures whose joint prior is described later in this section, and the kernel is given by

$$f_{\text{Beta}}(\cdot \mid a, b, T_0) := \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{1}{T_0} \left(\frac{t}{T_0}\right)^{a-1} \left(1 - \frac{t}{T_0}\right)^{b-1}, \qquad a > 0, b > 0, 0 < t < T_0.$$

The use of mixtures of Beta kernels allows the excitation function to have a flexible form, including non-monotonic decays. In fact, this type of mixtures have large support support of the space of absolutely continuous measures with bounded support on $[0, T_0]$ (see Section 4.3.1). The fact that the kernel has compact support also helps computationally without severely impacting the ability of the model to capture key features of the excitation function. Indeed, note that the compact support means that the number of observations pairs involved in the calculation of (4.1) and (4.2) grows linearly rather than quadratically with n. On the other hand, because the intensity function must eventually be strictly decreasing, in most cases a good approximation of intensity functions with infinite support can be obtained by choosing T_0 large enough. A similar approach, sometimes called tapering, is often used to speed up computation for Gaussian process models (e.g., see [55] and [82]).

Previous approaches to Bayesian nonparametric modeling of multivariate Hawkes process models (e.g., [42] and [102]) have relied on independent priors on each of the mixing distributions $G_{k,\ell}$. Instead, we propose to propose to model them jointly using further mixtures of the form

$$G_{k,\ell}(\cdot) = \varepsilon H_0(\cdot) + (1 - \varepsilon) H_{k,\ell}(\cdot), \qquad\qquad k, \ell = 1, \ldots, K,$$

where $0 \leq \varepsilon \leq 1$, and H_0 and the various $H_{k,\ell}$s are in turn given independent priors as described below. We can think of H_0 as a "common component" that captures features shared across all dimension pairs, and of $H_{k,\ell}$ as an "idiosyncratic" component that captures features that are specific to each. The parameter ε controls the relative weight associated

with each of these two components and, therefore, the level of dependence across the $G_{k,\ell}$s. Indeed, note that setting $\varepsilon = 0$ corresponds to assigning mutually independent priors to the $G_{k,\ell}$s, whereas $\varepsilon = 1$ implies that the intensity functions are exactly identical (perfect dependence) across all pairs. In practice, we expect ε to lie somewhere in between these two extremes, so we treat it as an unknown parameter that needs to be estimated from the data.

We model the common and the idiosyncratic components as independent realization from Dirichlet process priors, so that

$$H_0(\cdot) = \sum_{h=1}^{\infty} w_h^0 \delta_{(a_h^0, b_h^0)} \qquad\qquad H_{k,\ell}(\cdot) = \sum_{h=1}^{\infty} w_h^{k,\ell} \delta_{\left(a_h^{k,\ell}, b_h^{k,\ell}\right)}$$

where $\delta_{\boldsymbol{\theta}}(\cdot)$ denotes a point mass at $\boldsymbol{\theta}$, $a_h^0 \sim \text{Gamma}\left(c_a^C, d_a^C\right)$, $b_h^0 \sim \text{Gamma}\left(c_b^C, d_b^C\right)$, $a_h^{k,\ell} \sim \text{Gamma}\left(c_a^I, d_a^I\right)$ and $b_h^{k,\ell} \sim \text{Gamma}\left(c_b^I, d_b^I\right)$ independently for all $h = 1, 2, \ldots$ and $k, \ell = 1, \ldots, K$, and $w_h^0 = v_h^0 \prod_{r<h}(1 - v_h^0)$ and $w_h^{k,\ell} = v_h^{k,\ell} \prod_{r<h}(1 - v_h^{k,\ell})$, with $v_h^0 \sim \text{Beta}(\gamma, 1)$ and $v_h^{k,\ell} \sim \text{Beta}(\gamma, 1)$ also independent for all $h = 1, 2, \ldots$ and $k, \ell = 1, \ldots, K$. Allowing different parameters for the centering measure of the common and idiosyncratic components allows us to potentially encode relevant prior information. For example, it is common to assume that excitation functions are roughly decreasing, so we might want to reflect that in the shape of the common component by favoring values of $a_h^0 \ll 1$ and $b_h^0 \gg 1$. On the other hand, we might expect idiosyncratic components to reflect deviations from monotonicity, so we might want their centering measure to favor less extreme values for $a_h^{k,\ell}$ and $b_h^{k,\ell}$. Finally, and in a similar spirit, we model the mixture weight ϵ using a uniform distribution on $[0, 1]$.

It is worthwhile noting that the model in [115] has sometimes been criticized for being overparameterized. However, unlike other applications (e.g., see [140]), we are interested in estimating the aggregate excitation functions $\{\tilde{\phi}_{k,l}(t)\}$ defined in (4.3), which are themselves identifiable. This alleviates any concerns about overparameterization in this particular context.

The model is completed by eliciting priors for the parameters $\mu_1, \ldots, \mu_K$ and the coefficients $\{\alpha_{k,\ell}\}$ and the concentration γ. In particular, we set $\mu_\ell \sim \text{Gamma}\left(e_\ell, f_\ell\right)$ and $\alpha_{k,\ell} \sim \text{Gamma}\left(g_{k,\ell}, h_{k,\ell}\right)$ independently for all $k, \ell = 1, \ldots, K$.

4.3.1 Prior support

An important consideration when designing nonparametric priors is their associated support (e.g., see [160]). This question has been well studied for nonparametric mixture priors placed on a single unknown distribution, and they naturally extend to situations in which independent priors are assigned to individual members of a countable collection of distribution. However, results for dependent priors on collections of distribution have been less studied.

In order to show that our nonparametric prior has large Kullback–Leibler support, we discuss first a simpler result that applies to the individual priors we place on the common and idiosyncratic components:

Theorem 4.3.1. *Let $\mathcal{H}$ be the set of all absolutely continuous densities on $[0, T_0]$, and Π be the prior on $\mathcal{H}_0$ induced by a Dirichlet process mixture of the form $\int f_{Beta}(t \mid a, b, T_0) dH(a, b)$, where $H \sim DP(\gamma, \text{Gamma}\,(c_a, d_a) \times \text{Gamma}\,(c_b, d_b))$. Then, Π has full Kullback-Leibler support on $\mathcal{H}$, i.e., for any $\phi^0 \in \mathcal{H}$ and $\epsilon > 0$ we have*

$$\Pi\left(\left\{\phi : D_{KL}(\phi^0 \,||\, \phi) < \epsilon\right\}\right) > 0,$$

where $D_{KL}(\phi^0 \,||\, \phi) = \int \phi^0(s) \log(\phi^0(s)/\phi(s))\, \mathrm{d}s$ is the Kullback-Leibler divergence between ϕ^0 and ϕ.

Proof. The proof of the this theorem is a direct extension from the results in [163]. We show the KL property of f_0 by proving Conditions A1-A3 from Theorem 1 in [163] holds. Since we don't have ϕ in our case, Condition A2 is automatically satisfied. The remaining proof is similar to Theorem 11 in [163], except that the mixing distribution is drawn from a Dirichlet process, and that the two location parameters are also being mixed. For $\forall f_0 \in \mathcal{H}_0$ and $\forall \varepsilon > 0$, there exists a finite Beta mixture function f_{P_ε} with H mixtures where

$$f_{P_\varepsilon}(x) = \sum_{k=1}^{H} \frac{f_0\left(\frac{k-1}{H-1}\right)}{\sum_{k=1}^{H} f_0\left(\frac{k-1}{H-1}\right)} f_{\text{Beta}}(x \mid k, H-k) = \int f_{\text{Beta}}(x \mid a, b) dP_\varepsilon, \tag{4.4}$$

and

$$P_\varepsilon = \sum_{k=1}^{H} \omega_k \delta_{a,b}(a_k^0, b_k^0), \quad \omega_k^0 = \frac{f_0\left(\frac{k-1}{H-1}\right)}{\sum_{k=1}^{H} f_0\left(\frac{k-1}{H-1}\right)}, \quad a_k^0 = k, \quad b_k^0 = H - k, \tag{4.5}$$

such that

$$\int_0^1 f_0(x) \log \frac{f_0(x)}{f_{P_\varepsilon}(x)} dx < \varepsilon.$$

Thus Condition A1 holds. We then show Condition A3 also holds. First, we define $\mathcal{C}_\omega \subset \mathbb{S}^{H-1}, \mathcal{C}_{ab}^h \subset \mathbb{R}^2, h = 1, \ldots, H$ as sets such that

$$\mathcal{C}_\omega = \left\{ (\omega_1, \ldots, \omega_H) : \omega_h > \omega_h^0 e^{-\frac{\varepsilon}{4}}, \ \sum_{h=1}^H \omega_h = 1 \right\},$$

$$\mathcal{C}_{ab}^h = \big\{ (a_h, b_h) : a_h < a_h^0, b_h < b_h^0,$$
$$(a_h^0 - a_h) \left(\log x_M + \psi(a_h + b_h) - \psi(a_h) \right) + (b_h^0 - b_h) \left(\log(1 - x_M) + \psi(b_h + a_h) - \psi(b_h) \right) < \frac{\varepsilon}{4} \Big\}, \tag{4.6}$$

where

$$x_M = \max \left\{ d, 1 - d, \frac{a_h^0 - a_h}{a_h^0 - a_h + b_h^0 - b_h} \right\}.$$

Finally, we let $\mathcal{C}_{ab} = \oplus_{h=1}^H \mathcal{C}_{ab}^h$, $\boldsymbol{\omega} := (\omega_1, \ldots, \omega_H) \in \mathbb{S}^{H-1}, \boldsymbol{a} := (a_1, \ldots, a_H) \in \mathbb{R}^H, \boldsymbol{b} := (b_1, \ldots, b_H) \in \mathbb{R}^H$ and let $\mathcal{W} \subset \mathcal{H}$ be the set of finite mixture distributions induced by $\mathcal{C}_\omega$ and $\mathcal{C}_{ab}$:

$$\mathcal{W} := \left\{ P \in \mathcal{W} \mid P = \sum_{k=1}^H \omega_k \delta_{a,b} \left(a_k, b_k \right), \boldsymbol{\omega} \in \mathcal{C}_\omega, (a_h, b_h) \in \mathcal{C}_{ab}^h \right\}.$$

We note that for the chosen $\boldsymbol{\omega}_0, \boldsymbol{a}_0, \boldsymbol{b}_0$, for any $\omega \in \mathcal{C}_\omega$, we have

$$\frac{\sum_{j=1}^H \omega_h^0 f_{\text{Beta}} \left(x \mid a_h^0, b_h^0 \right)}{\sum_{j=1}^H \omega_h f_{\text{Beta}} \left(x \mid a_h^0, b_h^0 \right)} < e^{\frac{\varepsilon}{4}}, \quad \forall x \in (d, 1 - d). \tag{4.7}$$

We then note that

$$(a_h^0 - a_h) \log x + (b_h^0 - b_h) \log(1 - x) \leq (a_h^0 - a_h) \log x_M + (b_h^0 - b_h) \log(1 - x_M), \ \ \forall x \in (d, 1 - d). \tag{4.8}$$

Thus for any $(a_h, b_h) \in \mathcal{C}_{ab}^h$, consider the Cauchy remainder form of the first-order expansion

for $l(x, a_h^0, b_h^0) := \log f_{\text{Beta}}(x; a_h^0, b_h^0)$ at (a_h, b_h), we have

$$\begin{aligned} l(x; a_h^0, b_h^0) &= l(x; a_h, b_h) + \nabla l(x; a_h, b_h)^T \begin{bmatrix} a_h^0 - a_h \\ b_h^0 - b_h \end{bmatrix} + \frac{\theta^2}{2} \begin{bmatrix} a_h^0 - a_h & b_h^0 - b_h \end{bmatrix} \nabla^2 l(x; a_h^*, b_h^*) \begin{bmatrix} a_h^0 - a_h \\ b_h^0 - b_h \end{bmatrix} \\ &< l(x; a_h, b_h) + (a_h^0 - a_h)(\log x + \psi(a_h + b_h) - \psi(a_h)) + (b_h^0 - b_h)(\log(1-x) + \psi(b_h + a_h) - \psi(b_h)) \\ &< l(x; a_h, b_h) + (a_h^0 - a_h) \log x_M + (b_h^0 - b_h) \log(1 - x_M) \\ &< l(x; a_h, b_h) + \frac{\varepsilon}{4}, \quad \forall x \in (d, 1-d). \end{aligned} \tag{4.9}$$

Note that the chain of inequalities are based on equations (4.6) and (4.8), and the fact that

$$\nabla^2 l(x; a, b) \prec 0, a > 0, b > 0.$$

Finally, note that (4.9) is equivalent to

$$\frac{f_{\text{Beta}}(x \mid a_h^0, b_h^0)}{f_{\text{Beta}}(x \mid a_h, b_h)} < e^{\frac{\varepsilon}{4}}, \quad \forall x \in (d, 1-d), \tag{4.10}$$

and if for $h = 1, \ldots, H$, $(a_h, b_h) \in \mathcal{C}_{ab}^h$, for any $(\omega_1, \ldots, \omega_H) \in \mathcal{C}_\omega$, we have

$$\frac{\sum_{j=1}^H \omega_h f_{\text{Beta}}(x \mid a_h^0, b_h^0)}{\sum_{j=1}^H \omega_h f_{\text{Beta}}(x \mid a_h, b_h)} < e^{\frac{\varepsilon}{4}}, \quad \forall x \in (d, 1-d).$$

Combining equations (4.7) and (4.10), we have, for any $\boldsymbol{\omega} \in \mathcal{C}_\omega, (a_h, b_h) \in \mathcal{C}_{ab}^h, h = 1, \ldots, H$, we have

$$\begin{aligned} \frac{\sum_{j=1}^H \omega_h^0 f_{\text{Beta}}(x \mid a_h^0, b_h^0)}{\sum_{j=1}^H \omega_h f_{\text{Beta}}(x \mid a_h, b_h)} &= \frac{\sum_{j=1}^H \omega_h^0 f_{\text{Beta}}(x \mid a_h^0, b_h^0)}{\sum_{j=1}^H \omega_h f_{\text{Beta}}(x \mid a_h^0, b_h^0)} \\ &\times \frac{\sum_{j=1}^H \omega_h f_{\text{Beta}}(x \mid a_h^0, b_h^0)}{\sum_{j=1}^H \omega_h f_{\text{Beta}}(x \mid a_h, b_h)} = e^{\frac{\varepsilon}{2}}, \quad \forall x \in (d, 1-d). \end{aligned}$$

Thus

$$\int_d^{1-d} f_0(x) \log \frac{f_{P_\varepsilon}(x)}{f_P(x)} dx < \int_d^{1-d} f_0(x) dx \cdot \frac{\varepsilon}{2} < \frac{\varepsilon}{2}. \tag{4.11}$$

We then consider the scenario where $x \in (0, d] \bigcup [1-d, 1)$. We want to show that the likelihood ratio $\frac{f_{\text{Beta}}(x \mid a_h^0, b_h^0)}{f_{\text{Beta}}(x \mid a_h, b_h)}$ has a uniform finite upper bound over all $x \in (0, d] \bigcup [1-d, 1)$ and $(a_h, b_h) \in \mathcal{C}_{ab}^h, h = 1, \ldots, H$, i.e.

$$\begin{aligned} \sup_{x \in (0,d] \bigcup [1-d,1)} \frac{f_{\text{Beta}}(x \mid a_h^0, b_h^0)}{f_{\text{Beta}}(x \mid a_h, b_h)} &= \frac{\text{Be}(a_h, b_h)}{\text{Be}(a_h^0, b_h^0)} \sup_{x \in (0,d] \bigcup [1-d,1)} x^{a_h^0 - a_h} (1-x)^{b_h^0 - b_h} \\ &= \frac{\text{Be}(a_h, b_h)}{\text{Be}(a_h^0, b_h^0)} d^{a_h^0 + b_h^0 - a_h - b_h} \leq \frac{\text{Be}(a_h, b_h)}{\text{Be}(a_h^0, b_h^0)}, \end{aligned}$$

which follows from the fact that $a_h < a_h^0, b_h < b_h^0$. Thus we have

$$\sup_{\substack{\boldsymbol{a},\boldsymbol{b}\in\mathcal{C}_{ab}\\ \boldsymbol{\omega}\in\mathcal{C}_\omega}} \sup_{x\in(0,d]\bigcup[1-d,1)} \frac{\sum_{j=1}^H \omega_h^0 f_{\text{Beta}}\left(x \mid a_h^0, b_h^0\right)}{\sum_{j=1}^H \omega_h f_{\text{Beta}}\left(x \mid a_h, b_h\right)} \leq e^{-\frac{\varepsilon}{4}} \sup_{\boldsymbol{a},\boldsymbol{b}\in\mathcal{C}_{ab}} \frac{\text{Be}\left(a_h, b_h\right)}{\text{Be}\left(a_h^0, b_h^0\right)}$$
$$\leq \sup_{\boldsymbol{a},\boldsymbol{b}\in\mathcal{C}_{ab}} \frac{\text{Be}\left(a_h, b_h\right)}{\text{Be}\left(a_h^0, b_h^0\right)} := M < +\infty.$$

Thus we have

$$\int_0^d f_0(x) \log \frac{f_{P_\varepsilon}(x)}{f_P(x)} dx + \int_{1-d}^1 f_0(x) \log \frac{f_{P_\varepsilon}(x)}{f_P(x)} dx < M\left(F_0(d) + 1 - F_0(1-d)\right).$$

Thus, we can choose d small enough such that $M\left(F_0(d) + 1 - F_0(1-d)\right) < \frac{\varepsilon}{2}$, such that

$$\int_0^1 f_0(x) \log \frac{f_{P_\varepsilon}(x)}{f_P(x)} dx = \int_0^d f_0(x) \log \frac{f_{P_\varepsilon}(x)}{f_P(x)} dx + \int_{1-d}^1 f_0(x) \log \frac{f_{P_\varepsilon}(x)}{f_P(x)} dx + \int_d^{1-d} f_0(x)$$
$$< \frac{\varepsilon}{2} + \frac{\varepsilon}{4} + \frac{\varepsilon}{4} = \varepsilon.$$

Finally, as $\mathcal{C}_\omega, \mathcal{C}_{ab}$ are nonempty and open sets, we have $\Pi(\mathcal{W}) > 0$, thus $f_0 \in KL(\Pi)$. □

This result is the basis for the following corollary, which extends the result to the setting of the our dependent model:

Corollary 1. *Consider the joint prior* Π^* *on* $\mathcal{H}^* = \oplus_{k=1}^{K\times K}\mathcal{H}$ *induced by the nonparametric mixture of mixtures introduced in Section 4.3. Then, for any collection* $\{\phi_{1,1}^0, \ldots, \phi_{K,K}^0\} \in \mathcal{H}^*$ *and any* $\epsilon > 0$ *we have*

$$\Pi^*\left(\left\{\phi_{1,1}, \ldots, \phi_{K,K} : D_{KL}(\phi_{k,\ell}^0 \,||\, \phi_{k,\ell}) < \epsilon \text{ for all } k, \ell = 1, \ldots, K\right\}\right) > 0.$$

Proof. We can rewrite the setting in Corollary 1 as follows: consider $K \times K$ continuous densities on $[0,1]$, i.e. let $\phi_0^{1,1}, \ldots, \phi_0^{K,K} \in \mathcal{H}^*$. Consider the following joint prior $\Pi^* : \oplus_{h=1}^{K\times K}\mathcal{H}^* \to \mathbb{R}$ for $(\phi^{1,1}, \ldots, \phi^{K,K})$ such that

$$\begin{aligned}
\phi^{k,\ell} &= \delta g_0 + (1-\delta) g^{k,\ell}, k, \ell = 1, \ldots, K, \\
g^0(x) &= \int f_{\text{Beta}}(x \mid a, b) dP(a, b), \\
g^{k,\ell}(x) &= \int f_{\text{Beta}}(x \mid a, b) dP(a, b), k, \ell = 1, \ldots, K \qquad (4.12) \\
P &\sim \text{DP}(\gamma, \text{Gamma}(c_a, d_a) \times \text{Gamma}(c_b, d_b)), \\
\delta &\sim \text{Dirichlet}(1, 1),
\end{aligned}$$

To show this is true, let $\phi_0^0(x) := \frac{1}{K^2}\sum_{k=1}^K\sum_{\ell=1}^K \phi_0^{k,\ell}(x)$. For $\varepsilon > 0$, we let

$$\mathcal{G}^0 = \{g^0 \in \mathcal{H}_0 : \mathrm{KL}(\phi_0^0, g^0) < \frac{\varepsilon}{2}\}, \quad \mathcal{G}^{k,\ell} = \{g^{k,\ell} \in \mathcal{H}_0 : \mathrm{KL}(\phi_0^{k,\ell}, g^{k,\ell}) < \frac{\varepsilon}{2}\}.$$

Note that from Theorem 1.1 we have $\Pi_{g^0}(g^0 \in \mathcal{G}^0) > 0$ and $\Pi_{g^k}\left(g^{k,\ell} \in \mathcal{G}^{k,\ell}\right) > 0, k, \ell = 1, \ldots, K$. Let $M := \max_{1\le k,\ell\le M} \sup_{g_0\in\mathcal{G}_0} \mathrm{KL}(\phi_0^{k,\ell}, g^0) < +\infty$, we have, based on the convexity of KL divergence:

$$\mathrm{KL}(\phi_0^{k,\ell}, \delta g^0 + (1-\delta)g^{k,\ell}) \le \delta \mathrm{KL}(\phi_0^k, g^0) + (1-\delta)\mathrm{KL}(\phi_0^k, g^{k,\ell}) < \delta M + (1-\delta)\frac{\varepsilon}{2} < \varepsilon,$$

for all $\forall \delta < \frac{\varepsilon}{2M-\varepsilon}$, $g^0 \in \mathcal{G}^0$, $g^{k,\ell} \in \mathcal{G}^{k,\ell}, k, \ell = 1, \ldots, K$. Thus

$$\begin{aligned}
&\Pi^*\left(\left\{\phi^{1,1}, \ldots, \phi^{K,K} : \mathrm{KL}\left(\phi_0^k, \phi^k\right) < \varepsilon, k, \ell = 1, \ldots, K\right\}\right) \\
=&\Pi^*\left(\left\{g^0, g^1, \ldots, g^{K,K}, \delta : \mathrm{KL}\left(\phi_0^k, \delta g^0 + (1-\delta)g^k\right) < \varepsilon, k, \ell = 1, \ldots, K\right\}\right) \\
>&\Pi^*\left(\left\{g^0, g^1, \ldots, g^{K,K}, \delta : g^0 \in \mathcal{G}^0, g^k \in \mathcal{G}^{k,\ell}, k, \ell = 1, \ldots, K, \delta < \frac{\varepsilon}{2M-\varepsilon}\right\}\right) \\
=&\Pi_{g^0}\left(g^0 \in \mathcal{G}^0\right) \prod_{1\le k,\ell\le K} \Pi_{g^{k,\ell}}(g^{k,\ell} \in \mathcal{G}^{k,\ell})\Pi_\delta(\delta < \frac{\varepsilon}{2M-\varepsilon}) > 0.
\end{aligned}$$

□

4.4 Computation

This Section describes two alternative computational strategies for inference on the posterior distribution associated with the model described in Section 4.2. The first strategy is based on a Markov chain Monte Carlo (MCMC) algorithm that heavily relies on Gibbs sampling steps. This algorithm is relatively simple to implement and can provide accurate estimates of most posterior quantities of interest, but it is comparatively slow and impractical for large datasets. The second strategy we discuss is based on a stochastic gradient variational approximation to the posterior distribution. The resulting algorithm is orders of magnitudes faster than MCMC and tends to yield reasonably accurate point estimates, but it also tends to underestimates the uncertainty associated with the posterior distribution.

Both of the computational strategies we discuss rely on a common set of approximations and latent variable augmentations, which we describe before getting into the details of each of them. Firstly, both algorithms rely on (an approximation to) the full data likelihood in

(4.2). Augmenting the model with the (latent) matrix $\mathbf{B}$ is what enables us to treat the problem of estimating the excitation function as one of density estimation, which in turn serves as an additional motivation for the use of mixture models in Section 4.2. The main challenge with this approach is that (3.3) is not fully tractable. In particular, note that (3.3) includes a term of the form

$$\Upsilon = \sum_{\ell=1}^{K} \int_0^T \lambda_\ell(s)ds = \sum_{\ell=1}^{K} \mu_\ell T + \sum_{k=1}^{K}\sum_{\ell=1}^{K} \alpha_{k,\ell} \sum_{d_i=k} \tilde{\Phi}_{k,\ell}(T - t_i). \tag{4.13}$$

The term $\int_0^T \lambda_\ell(s)ds$ is often called the *compensator* for the conditional density function $\lambda_\ell(t)$, and it captures the likelihood of there being infinitely many none-events between observations on the temporal domain $[0, T_0]$. The complex structure associated with the compensator can a major hurdle in the development of computational algorithms for Hawkes process models (e.g., see [79]). A common solution, which we adopt in this chapter, is the approximation suggested in [90]:

$$\alpha_{k,\ell}\tilde{\Phi}_{k,\ell}(T - t_i) \approx \alpha_{k,\ell}. \tag{4.14}$$

This approximation is particularly appealing in our context because we assume that $\phi_{k,\ell}(\cdot)$ has bounded support over $[0, T_0]$. Therefore, the approximation will only affect the likelihood evaluation for a relative small portion of the whole observations.

Secondly, for the purposes of computational implementation, we consider a finite truncation approximation on the number of component in our model that relies on the ideas of [75]. More specifically, we consider the finite mixture model of the form

$$\tilde{\phi}_{k,\ell}(t) = \varepsilon \sum_{h=1}^{H} p_h^0 f_{\text{Beta}}(t \mid a_h^0, b_h^0) + (1-\varepsilon) \sum_{h=1}^{H} p_h^{k,\ell} f_{\text{Beta}}(t \mid a_h^{k,\ell}, b_h^{k,\ell}),$$

where

$$\boldsymbol{p}^0 = (p_1^0, \ldots, p_H^0) \sim \text{Dirichlet}\left(\frac{\gamma}{H}, \ldots, \frac{\gamma}{H}\right), \quad \boldsymbol{p}^{k,l} = (p_1^{k,\ell}, \ldots, p_H^{k,\ell}) \sim \text{Dirichlet}\left(\frac{\gamma}{H}, \ldots, \frac{\gamma}{H}\right).$$

for all $k, \ell = 1, \ldots, K$. The truncation level H is an upper bound on the number of mixture components. As long as H is chosen to be large enough, this kind of (overfitted) finite-dimensional mixture provides an excellent approximation to the its nonparametric

counterpart, one that has been repeatedly exploited to design computational algorithms for non-parametric mixture models.

Finally, and as is common in the context of mixture modeling, we introduce various sets of latent allocation variables, indicating the mixture component each observation comes from. Since (4.2) is evaluated on inter-arrival times across dimensions, we define

$$\mathcal{T}_{k,\ell} = \{(i,j) : d_i = k, d_j = \ell, i < j\}$$

as the set of all tuples of observation indices corresponding to transitions between a potential parent in dimension i and a potential offspring in dimension j. Note that we only evaluate the transitions indicated by the branching structure, i.e. $B_{ij} = 1$. For each tuple $(i,j) \in \mathcal{T}_{k,\ell}$, we denote $(W_{i,j}, Z_{i,j})$ as a set of latent variables that jointly determine the mixture component that $t_j - t_i$ belongs to. More specifically,

$$t_j - t_i \mid Z_{i,j}, W_{i,j} \sim \begin{cases} \text{Beta}\left(a^0_{Z_{i,j}}, b^0_{Z_{i,j}}\right) & \text{if } W_{i,j} = 1 \\ \text{Beta}\left(a^{k,\ell}_{Z_{i,j}}, b^{k,\ell}_{Z_{i,j}}\right) & \text{if } W_{i,j} = 0 \end{cases}.$$

Given $\boldsymbol{p}^0$, $\{\boldsymbol{p}^{k,\ell}\}$ and ϵ, $\{W^{k,\ell}_{i,j}\}$ and $\{Z^{k,\ell}_{i,j}\}$ haver a joint distribution given by:

$$P(Z_{i,j} = h, W_{i,j} = 1 \mid \boldsymbol{p}^0, \varepsilon) = \varepsilon p^0_h,$$
$$P(Z_{i,j} = h, W_{i,j} = 0 \mid \{\boldsymbol{p}^{k,\ell}\}, \varepsilon) = (1-\varepsilon) p^{k,\ell}_h.$$

4.4.1 Markov chain Monte Carlo algorithm

Once the approximations and data augmentation discussed above are introduced, the posterior distribution takes the form

$$\begin{aligned} f\left(\{\mu_k\}, \{\alpha_{k,\ell}\}, \{W_{i,j}\}, \{Z_{i,j}\}, \{p^0_h\}, \{a^0_h\}, \{b^0_h\}, \{p^{k,\ell}_h\}, \{a^{k,\ell}_h\}, \{b^{k,\ell}_h\}, \varepsilon, \boldsymbol{B} \mid \boldsymbol{X}\right) \propto \\ f^*\left(\boldsymbol{X}, \boldsymbol{B} \mid \{\mu_k\}, \{\alpha_{k,\ell}\}, \{W_{i,j}\}, \{Z_{i,j}\}, \{a^0_h\}, \{b^0_h\}, \{a^{k,\ell}_h\}, \{b^{k,\ell}_h\}\right) \\ f\left(\{\mu_k\}, \{\alpha_{k,\ell}\}, \{W_{i,j}\}, \{Z_{i,j}\}, \{a^0_h\}, \{b^0_h\}, \{a^{k,\ell}_h\}, \{b^{k,\ell}_h\}\right) \end{aligned} \quad (4.15)$$

where

$$\log f^*\left(\boldsymbol{X}, \boldsymbol{B} \mid \{\mu_k\}, \{\alpha_{k,\ell}\}, \{W_{i,j}\}, \{Z_{i,j}\}, \{a_h^0\}, \{b_h^0\}, \{a_h^{k,\ell}\}, \{b_h^{k,\ell}\}\right) = \sum_{k=1}^{K}\sum_{\ell=1}^{K} |O_{k,\ell}| \left(\log \alpha_{k,\ell}\right)$$
$$\sum_{\substack{\{(i,j):i<j \\ d_i=k, d_j=\ell\}}} B_{i,j} I(Z_{i,j}=h)\Big[W_{i,j} \log f_{\text{Beta}}\left(t_j - t_i \mid a_h^{k,\ell}, b_h^{k,\ell}\right)$$
$$+ (1 - W_{i,j}) \log f_{\text{Beta}}\left(t_j - t_i \mid a_h^0, b_h^0\right)\Big] - \alpha_{k,\ell} n_k\Big],$$

with $n_k = \sum_{i=1}^{N} I(d_i = k)$, is the approximate log-likelihood based on (4.14), and the joint prior introduced above defines $f\left(\{\mu_k\}, \{\alpha_{k,\ell}\}, \{W_{i,j}\}, \{Z_{i,j}\}, \{a_h^0\}, \{b_h^0\}, \{a_h^{k,\ell}\}, \{b_h^{k,\ell}\}\right)$.

Many of the full conditional distributions associated with (4.15) belong to known families of distributions and are therefore easy to sample from. For example, the full conditional posteriors for each of the μ_ks and $\alpha_{k,\ell}$s correspond to independent Gamma distributions. Similarly the branching structure $\boldsymbol{B}$ and the indicators $\{Z_{i,j}\}$ and $\{W_{i,j}\}$ are all categorical variables and sampling from their full conditional distributions is straightforward. Finally, the full conditional posterior distribution for $\boldsymbol{p}^0$ and each of the $\boldsymbol{p}^{k,l}$s correspond to Dirichlet distributions on the H dimensional simplex, and the full conditional for ε follows an updated Beta distribution. The key exception are the parameters $\{a_h^0\}$, $\{b_h^0\}$, $\{a_h^{k,\ell}\}$ and $\{b_h^{k,\ell}\}$. We propose to update these though a random walk Metropolis-within-Gibbs steps on their log scales. Details of the algorithm can be seen in Section 4.4.3.

4.4.2 Stochastic variational inference

In variational inference, the intractable posterior distribution $f(\boldsymbol{\theta} \mid \boldsymbol{X})$ is replaced with a tractable approximation $q_{\tilde{\boldsymbol{\eta}}}(\boldsymbol{\theta})$ obtained by minimizing the Kullback–Leibler divergence between it and a member of a carefully chosen parametric family $\{q_{\boldsymbol{\eta}}(\boldsymbol{\theta}) : \boldsymbol{\eta} \in H\}$, i.e.,

$$\tilde{\boldsymbol{\eta}} = \arg\max_{\boldsymbol{\eta} \in H} \mathrm{E}_{q_{\boldsymbol{\eta}}} \log\left\{\frac{p(\boldsymbol{\theta}, \mathbf{X})}{q_{\boldsymbol{\eta}}(\boldsymbol{\theta})}\right\} := \arg\max_{\boldsymbol{\eta} \in H} \mathrm{ELBO}_{\boldsymbol{\eta}}\,.$$

In the sequel, we work with a mean-field variational approximation that assumes inde-

pendence across all parameters, except for the $(W_{i,j}, Z_{i,j})$ pairs,

$$q_{\boldsymbol{\eta}}\left(\{\mu_k\},\{\alpha_{k,\ell}\},\{W_{i,j}\},\{Z_{i,j}\},\{p_h^0\},\{a_h^0\},\{b_h^0\},\{p_h^{k,\ell}\},\{a_h^{k,\ell}\},\{b_h^{k,\ell}\},\varepsilon,\boldsymbol{B} \mid \boldsymbol{X}\right) =$$
$$\prod_{\ell=1}^{K} q_{\eta_{\mu_\ell}}(\mu_\ell) \prod_{k=1}^{K}\prod_{\ell=1}^{K} q_{\eta_{\alpha_{k,\ell}}}(\alpha_{k,\ell}) \prod_{h=1}^{L_0} q_{\eta_{a_h^0}}(a_h^0) q_{\eta_{b_h^0}}(b_h^0) q_{\eta_{p_h^0}}(p_h^0) \prod_{k=1}^{K}\prod_{\ell=1}^{K}\prod_{h=1}^{L} q_{\eta_{a_h^{k,\ell}}}(a_h^{k,\ell}) q_{\eta_{b_h^{k,\ell}}}(b_h^{k,\ell}) q_{\eta_{p_h^{k,\ell}}}(p_h^{k,\ell})$$
$$\prod_{j=1}^{N}\prod_{i=1}^{j} q_{\eta_{B_{ij}}}(B_{ij}) \prod_{k=1}^{K}\prod_{\ell=1}^{K} \prod_{(i,j)\in\mathcal{T}_{k,l}} q_{\eta_{W_{i,j}},\eta_{Z_{i,j}}}(W_{i,j}, Z_{i,j})\, q_{\eta_\varepsilon}(\varepsilon). \quad (4.16)$$

The families to which each of the individual terms in the variational approximation belong match those of the corresponding priors, facilitating computation (e.g., see [16]). Indeed, similarly to the MCMC algorithm, the conditional conjugacy of many of the priors and the choice of the approximation family means that most of the parameters can be easily optimized using an iterative coordinate descent algorithm that alternates updates of each of the parameters. Again, the one exception are the parameters $\{a_h^0\}$, $\{b_h^0\}$, $\{a_h^{k,\ell}\}$ and $\{b_h^{k,\ell}\}$. To update them, we rely on the approximate lower bound for the ELBO that was introduced in [99]. More specifically, we approximate $\mathrm{E}_{a,b}\left[\log \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\right]$ using a Taylor's expansion:

$$\begin{aligned}
\mathrm{E}_{a,b}\left[\log \frac{\Gamma(a+b)}{\Gamma(a)\,\Gamma(b)}\right] \geq \log \frac{\Gamma\left(\bar{a}+\bar{b}\right)}{\Gamma\left(\bar{a}\right)\Gamma\left(\bar{b}\right)} &+ \bar{a}[\psi(\bar{a}+\bar{b}) - \psi(\bar{a})](\mathrm{E}[\log a] - \log \bar{a}) \\
&+ \bar{b}[\psi(\bar{a}+\bar{b}) - \psi(\bar{b})](\mathrm{E}[\log b] - \log \bar{b}) \\
&+ \frac{1}{2}\bar{a}^2\left[\psi'(\bar{a}+\bar{b}) - \psi'(\bar{a})\right]\mathrm{E}\left[(\log a - \log \bar{a})^2\right] \\
&+ \frac{1}{2}\bar{b}^2\left[\psi'(\bar{a}+\bar{b}) - \psi'(\bar{b})\right]\mathrm{E}\left[(\log b - \log \bar{b})^2\right] \\
&+ \bar{a}\cdot\bar{b}\cdot\psi'(\bar{a}+\bar{b})(\mathrm{E}[\log a] - \log \bar{b})(\mathbf{E}[\log b] - \log \bar{b}).
\end{aligned} \quad (4.17)$$

where $\psi(z) = \frac{\mathrm{d}}{\mathrm{d}z}\ln\Gamma(z) = \frac{\Gamma'(z)}{\Gamma(z)}$ denotes the Digamma function [2], and $\bar{a}, \bar{b}$ are the expectations of a and b under the variational Gamma distribution, e.g. $\bar{a} = \frac{\eta_{a,1}}{\eta_{a,2}}$ for $a \sim \mathrm{Gamma}(\eta_{a,1}, \eta_{a,2})$. The use of this lower bound leads to simple, closed form updates for $\boldsymbol{\eta}_{a^0}$s, $\boldsymbol{\eta}_{b^0}$s, $\boldsymbol{\eta}_{a^{k,\ell}}$s and $\boldsymbol{\eta}_{b^{k,\ell}}$s.

Finally, to further speed up computation, we implement a stochastic gradient version of the conjugate descent algorithm that closely follows the ideas of [70] (see [79] as well). At each iteration, the algorithm updates the allocation and branching variables $W_{i,j}$, $Z_{i,j}$ and $B_{i,j}$ for the observations contained in a randomly chosen segment of length κ within the

interval $[0, T]$ before updating the remainder of the model parameters. The learning rate we use for this update is given by $\rho_s = \rho_0(s+\tau_1)^{\tau_2}$, where ρ_0 is a common scaling factor, $\tau_1 \geq 0$ is the delay parameter that 'slows down' early iterations and $\tau_2 \in (0.5, 1]$ is the forgetting rate that controls the rate of the exponential decay.

Computing ELBO

Under our modeling assumptions, the ELBO can be expressed according to the following formula:

$$
\begin{aligned}
\mathrm{ELBO}_{\boldsymbol{\eta}} = & \sum_{\ell=1}^{K} \mathrm{ELBO}_{\boldsymbol{\eta}}(\mu_\ell) + \sum_{k=1}^{K}\sum_{\ell=1}^{K} \mathrm{ELBO}_{\boldsymbol{\eta}}(\alpha_{k,\ell}) + \sum_{h=1}^{L_0} \mathrm{ELBO}_{\boldsymbol{\eta}}\left(p_h^0\right) + \mathrm{ELBO}_{\eta}\left(a_h^0\right) + \mathrm{ELBO}_{\eta}\left(b_h^0\right) \\
& + \sum_{k=1}^{K}\sum_{\ell=1}^{K}\sum_{h=1}^{L} \mathrm{ELBO}_{\boldsymbol{\eta}}\left(p_h^{k,\ell}\right) + \mathrm{ELBO}_{\eta}\left(a_h^{k,\ell}\right) + \mathrm{ELBO}_{\eta}\left(b_h^{k,\ell}\right) + \mathrm{ELBO}_{\boldsymbol{\eta}}(\varepsilon) \\
& + \sum_{k=1}^{K}\sum_{\ell=1}^{K} \sum_{(i,j)\in\mathcal{T}_{k,\ell}} \eta_{B_{i,j}} \log \frac{T_b}{(t_i - t_j)(T_b - (t_i - t_j))} \\
& + \sum_{h=1}^{L_0} \mathrm{E}_{q_{\eta_{a_h^0, b_h^0}}} \left[\log \frac{\Gamma\left(a_h^0 + b_h^0\right)}{\Gamma\left(a_h^0\right)\Gamma\left(b_h^0\right)} \right] \mathrm{E}_{q_{\eta_{\mathrm{B,w,z}}}} \left[N_h^0\right] \\
& + \sum_{k=1}^{K}\sum_{\ell=1}^{K}\sum_{h=1}^{L} \left[\log \frac{\Gamma\left(a_h^{k,\ell} + b_h^{k,\ell}\right)}{\Gamma\left(a_h^{k,\ell}\right)\Gamma\left(b_h^{k,\ell}\right)} \right] \mathrm{E}_{q_{\eta_{\mathrm{B,w,z}}}} \left[N_h^{k,\ell}\right],
\end{aligned}
\tag{4.18}
$$

where

$$\begin{aligned}
\text{ELBO}_{\boldsymbol{\eta}}(\mu_\ell) &= \left[\psi\left(\eta_{\mu,\ell,1}\right) - \log\left(\eta_{\mu,\ell,2}\right)\right]\left(\text{E}_{q_{\eta_\text{B}}}\left[|I_\ell|\right] + a_\mu - \eta_{\mu,\ell,1}\right) - \frac{\eta_{\mu,\ell,1}}{\eta_{\mu,\ell,2}}(T + b_\mu - \eta_{\mu,\ell,2}) \\
\text{ELBO}_{\boldsymbol{\eta}}(\alpha_{k,\ell}) &= \left[\psi\left(\eta_{\alpha,k,\ell,1}\right) - \log\left(\eta_{\alpha,k,\ell,2}\right)\right]\left(\text{E}_{q_{\eta_\text{B}}}\left[|O_{k,\ell}|\right] + e_{k,\ell} - \eta_{\alpha,k,\ell,1}\right) - \frac{\eta_{\alpha,k,\ell,1}}{\eta_{\alpha,k,\ell,2}}\left(n_k + f_{k,\ell} - \eta_{\alpha,k,\ell,2}\right) \\
\text{ELBO}_{\boldsymbol{\eta}}(p_h^0) &= \left[\psi\left(\eta_{p,0,h}\right) - \psi\left(\sum_{h=1}^{L_0}\eta_{p,0,h}\right)\right]\left(\text{E}_{q_{\eta_{\mathbf{B},\mathbf{w},\mathbf{z}}}}\left[N_h^0\right] + \frac{\alpha_{DP}}{L_0} - \eta_{p,0,j}\right) \\
\text{ELBO}_{\boldsymbol{\eta}}(p_h^{k,\ell}) &= \left[\psi\left(\eta_{p,k,\ell,h}\right) - \psi\left(\sum_{h=1}^{L}\eta_{p,k,\ell,h}\right)\right]\left(\text{E}_{q_{\eta_{\mathbf{B},\mathbf{w},\mathbf{z}}}}\left[N_h^{k,\ell}\right] + \frac{\alpha_{DP}}{L} - \eta_{p,k,\ell,j}\right) \\
\text{ELBO}_{\boldsymbol{\eta}}(\varepsilon) &= \left[\psi\left(\eta_\varepsilon\right) - \psi\left(1\right)\right]\left(\sum_{h=1}^{L_0}\text{E}_{q_{\eta_{\mathbf{B},\mathbf{w},\mathbf{z}}}}\left[N_h^0\right] + 1 - \eta_\varepsilon\right) \\
&\quad + \left[\psi\left(1-\eta_\varepsilon\right) - \psi\left(1\right)\right]\left(\sum_{k=1}^{K}\sum_{\ell=1}^{K}\sum_{h=1}^{L}\text{E}_{q_{\eta_{\mathbf{B},\mathbf{w},\mathbf{z}}}}\left[N_h^{k,\ell}\right] + 1 - \eta_\varepsilon\right) \\
\text{ELBO}_{\boldsymbol{\eta}}(a_h^{k,\ell}) &= \frac{\eta_{a,k,\ell,h,1}}{\eta_{a,k,\ell,h,2}}\left[\sum_{(i,j)\in\mathcal{T}_{k,\ell}}\eta_{B_{i,j}}\eta_{W_{i,j}}\eta_{Z_{i,j,h}}\log\left(\frac{t_i - t_j}{T_b}\right) + d_a - \eta_{a,k,\ell,h,2}\right] \\
\text{ELBO}_{\boldsymbol{\eta}}(b_h^{k,\ell}) &= \frac{\eta_{b,k,\ell,h,1}}{\eta_{b,k,\ell,h,2}}\left[\sum_{(i,j)\in\mathcal{T}_{k,\ell}}\eta_{B_{i,j}}\eta_{W_{i,j}}\eta_{Z_{i,j,h}}\log\left(1 - \frac{t_i - t_j}{T_b}\right) + d_b - \eta_{b,k,\ell,h,2}\right] \\
\text{ELBO}_{\boldsymbol{\eta}}(a_h^0) &= \frac{\eta_{a,0,h,1}}{\eta_{a,0,h,2}}\left[\sum_{k=1}^{K}\sum_{\ell=1}^{K}\sum_{(i,j)\in\mathcal{T}_{k,\ell}}\eta_{B_{i,j}}\left(1-\eta_{W_{i,j}}\right)\eta_{Z_{i,j,h}}\log\left(\frac{t_i - t_j}{T_b}\right) + d_a - \eta_{a,0,h,2}\right] \\
\text{ELBO}_{\boldsymbol{\eta}}(b_h^0) &= \frac{\eta_{b,0,h,1}}{\eta_{b,0,h,2}}\left[\sum_{k=1}^{K}\sum_{\ell=1}^{K}\sum_{(i,j)\in\mathcal{T}_{k,\ell}}\eta_{B_{i,j}}\left(1-\eta_{W_{i,j}}\right)\eta_{Z_{i,j,h}}\log\left(1 - \frac{t_i - t_j}{T_b}\right) + d_b - \eta_{b,0,h,2}\right].
\end{aligned}$$

Updating $\boldsymbol{\eta}_{\mu_p^0}, \boldsymbol{\eta}_{\mu_p^{k,\ell}}$

$$\begin{aligned}
\eta_{p,0,j}^{(r+1)} &= (1-\rho_r)\eta_{p,0,j}^{(r)} + \rho_r\left(\frac{\alpha_{DP}}{L_0} + \sum_{k=1}^{K}\sum_{\ell=1}^{L}\sum_{(i,j)\in\mathcal{T}_{k,i}}\eta_{B_{i,j}}^{(r)}\eta_{W_{i,j}}^{(r)}\eta_{Z_{i,j,h,0}}^{(r)}.\right) \\
\eta_{p,k,\ell,j}^{(r+1)} &= (1-\rho_r)\eta_{p,k,\ell,j}^{(r)} + \rho_r\left(\frac{\alpha_{DP}}{L_0} + \sum_{(i,j)\in\mathcal{T}_{\mathcal{T},\boldsymbol{i}}}\eta_{B_{i,j}}^{(r)}\left(1-\eta_{W_{i,j}}^{(r)}\right)\eta_{Z_{i,j,h,1}}^{(r)}.\right)
\end{aligned}$$

Updating $\boldsymbol{\eta}_{\varepsilon}$

$$\eta_{\varepsilon,1}^{(r+1)} = (1-\rho_r)\eta_{\varepsilon,1}^{(r)} + \rho_r \left(1 + \sum_{h=1}^{L_0}\sum_{k=1}^{K}\sum_{\ell=1}^{L}\sum_{(i,j)\in\mathcal{T}_{k,i}} \eta_{B_{i,j}}^{(r)}\eta_{W_{i,j}}^{(r)}\eta_{Z_{i,j,h,0}}^{(r)}\right)$$

$$\eta_{\varepsilon,2}^{(r+1)} = (1-\rho_r)\eta_{\varepsilon,2}^{(r)} + \rho_r \left(1 + \sum_{k=1}^{K}\sum_{\ell=1}^{K}\sum_{h=1}^{L}\sum_{(i,j)\in\mathcal{T}_{\mathcal{T},i}} \eta_{B_{i,j}}^{(r)}\left(1-\eta_{W_{i,j}}^{(r)}\right)\eta_{Z_{i,j,h,1}}^{(r)}\right)$$

Updating $\boldsymbol{\eta}_{\mu_\ell}, \boldsymbol{\eta}_{\alpha_{k,\ell}}$

$$\begin{aligned}
\eta_{\mu,\ell,1}^{(r+1)} &= (1-\rho_r)\,\eta_{\mu,\ell,1}^{(r)} + \rho_r\left(a_\mu + \sum_{i=1}^{N}\eta_{B_{i,i}}^{(r)}\right)\\
\eta_{\mu,\ell,2}^{(r+1)} &= (1-\rho_r)\,\eta_{\mu,\ell,1}^{(r)} + \rho_r\,(b_\mu + T)\\
\eta_{\alpha,k,\ell,1}^{(r+1)} &= (1-\rho_r)\eta_{\alpha,k,\ell,1}^{(r)} + \rho_r\left(e_{k,\ell} + \sum_{d_i=k,d_j=\ell,i<j}\eta_{B_{i,j}}^{(r)}\right)\\
\eta_{\alpha,k,\ell,2}^{(r+1)} &= (1-\rho_r)\eta_{\alpha,k,\ell,2}^{(r)} + \rho_r\,(n_k + f_{k,\ell})
\end{aligned}$$

Updating $\boldsymbol{\eta}_{a_{k,\ell,h}}, \boldsymbol{\eta}_{b_{k,\ell,h}}$

Applying the lower bound approximation using (8) in the manuscript, we obtain the conjugate structure for $a_h^{k,\ell}$:

$$\begin{aligned}
\text{ELBO}_{\boldsymbol{\eta}}\left(a_h^{k,\ell}\right) \geq\ & \left[\psi\left(\eta_{a,k,\ell,h,1}\right) - \log\left(\eta_{a,k,\ell,h,2}\right)\right]\Big\{\Big[\bar{a}_{k,\ell,h}\left(\psi\left(\bar{a}_{k,\ell,h}+\bar{b}_{k,\ell,h}\right) - \psi\left(\bar{a}_{k,\ell,h}\right)\right)\\
&+\bar{a}_{k,\ell,h}\bar{b}_{k,\ell,h}\psi'\left(\bar{a}_{k,\ell,h}+\bar{b}_{k,\ell,h}\right)\left(\psi\left(\eta_{b,k,\ell,h,1}\right) - \log\left(\eta_{b,k,\ell,h,2}\right) - \log\bar{b}_{k,\ell,h}\right)\Big]\\
&\times \mathrm{E}_{q_{\eta_{\mathbf{B},\mathbf{w},\mathbf{z}}}}\left[N_h^{k,\ell}\right] + c_a - \eta_{a,k,\ell,h,1}\Big\}\\
&+\frac{\eta_{a,k,\ell,h,1}}{\eta_{a,k,\ell,h,2}}\left[\sum_{(i,j)\in\mathcal{T}_{k,\ell}}\eta_{B_{i,j}}\eta_{W_{i,j}}\eta_{Z_{i,y,h}}\log\left(\frac{t_i^{(r)}-t_j^{(r)}}{T_b}\right) + d_a - \eta_{a,k,\ell,h,2}\right]
\end{aligned}$$

$$
\begin{aligned}
\mathrm{ELBO}_{\boldsymbol{\eta}}\left(b_h^{k,\ell}\right) \geq & \left[\psi\left(\eta_{b,k,\ell,h,1}\right)-\log\left(\eta_{b,k,\ell,h,2}\right)\right]\left\{\left[\bar{b}_{k,\ell,h}\left(\psi\left(\bar{a}_{k,\ell,h}+\bar{b}_{k,\ell,h}\right)-\psi\left(\bar{b}_{k,\ell,h}\right)\right)\right.\right. \\
& +\bar{a}_{k,\ell,h}\bar{b}_{k,\ell,h}\psi'\left(\bar{a}_{k,\ell,h}+\bar{b}_{k,\ell,h}\right)\left(\psi\left(\eta_{a,k,\ell,h,1}\right)-\log\left(\eta_{a,k,\ell,h,2}\right)-\log\bar{a}_{k,\ell,h}\right)\Big] \\
& \left.\times \mathrm{E}_{q_{\eta_{\mathbf{B},\mathbf{w},\mathbf{z}}}}\left[N_h^{k,\ell}\right]+c_b-\eta_{b,k,\ell,h,1}\right\} \\
& +\frac{\eta_{b,k,\ell,h,1}}{\eta_{b,k,\ell,h,2}}\left[\sum_{(i,j)\in\mathcal{T}_{k,\ell}}\eta_{B_{i,j}}\eta_{W_{i,j}}\eta_{Z_{i,y,h}}\log\left(1-\frac{t_i^{(r)}-t_j^{(r)}}{T_b}\right)+d_b-\eta_{b,k,\ell,h,2}\right]
\end{aligned}
$$

We update $\eta^{(r)}_{a,k,\ell,h,1}, \eta^{(r)}_{a,k,\ell,h,2}$ by the following formula:

$$
\begin{aligned}
\eta^{(r+1)}_{a,k,\ell,h,1} = & (1-\rho_r)\eta^{(r)}_{a,k,\ell,h,1}+\rho_r\left\{\left[\frac{\eta^{(r)}_{a,k,\ell,h,1}}{\eta^{(r)}_{a,k,\ell,h,2}}\left(\psi\left(\frac{\eta^{(r)}_{a,k,\ell,h,1}}{\eta^{(r)}_{a,k,\ell,h,2}}+\frac{\eta^{(r)}_{b,k,\ell,h,1}}{\eta^{(r)}_{b,k,\ell,h,2}}\right)-\psi\left(\frac{\eta^{(r)}_{a,k,\ell,h,1}}{\eta^{(r)}_{a,k,\ell,h,2}}\right)\right)\right.\right. \\
& \left.+\frac{\eta^{(r)}_{a,k,\ell,h,1}}{\eta^{(r)}_{a,k,\ell,h,2}}\frac{\eta^{(r)}_{b,k,\ell,h,1}}{\eta^{(r)}_{b,k,\ell,h,2}}\psi'\left(\frac{\eta^{(r)}_{a,k,\ell,h,1}}{\eta^{(r)}_{a,k,\ell,h,2}}+\frac{\eta^{(r)}_{b,k,\ell,h,1}}{\eta^{(r)}_{b,k,\ell,h,2}}\right)\left(\psi\left(\eta^{(r)}_{b,k,\ell,h,1}\right)-\log\left(\eta^{(r)}_{b,k,\ell,h,2}\right)-\log\frac{\eta^{(r)}_{b,k,\ell,h,1}}{\eta^{(r)}_{b,k,\ell,h,2}}\right)\right] \\
& \left.\times\kappa^{-1}\sum_{(i,j)\in\mathcal{T}_{k,\ell}}\eta^{(r)}_{B_{i,j}}\left(1-\eta^{(r)}_{W_{i,j}}\right)\eta^{(r)}_{Z_{i,j}}+c_a\right\} \\
\eta^{(r+1)}_{a,k,\ell,h,2} = & (1-\rho_r)\eta^{(r)}_{a,k,\ell,h,2}+\rho_r\left\{\kappa^{-1}\sum_{(i,j)\in\mathcal{T}_{k,\ell}}\eta_{B^{(r)}_{i,j}}\eta_{W^{(r)}_{i,j}}\eta_{Z^{(r)}_{i,j,h}}\log\left(\frac{t_i^{(r)}-t_j^{(r)}}{T_b}\right)+d_a\right\}.
\end{aligned}
\tag{4.19}
$$

Similarly, we can update $\eta^{(r)}_{b,k,\ell,h,1}, \eta^{(r)}_{b,k,\ell,h,2}$ as follows:

$$
\begin{aligned}
\eta^{(r+1)}_{b,k,\ell,h,1} = & (1-\rho_r)\eta^{(r)}_{b,k,\ell,h,1}+\rho_r\left\{\left[\frac{\eta^{(r)}_{b,k,\ell,h,1}}{\eta^{(r)}_{b,k,\ell,h,2}}\left(\psi\left(\frac{\eta^{(r)}_{a,k,\ell,h,1}}{\eta^{(r)}_{a,k,\ell,h,2}}+\frac{\eta^{(r)}_{b,k,\ell,h,1}}{\eta^{(r)}_{b,k,\ell,h,2}}\right)-\psi\left(\frac{\eta^{(r)}_{b,k,\ell,h,1}}{\eta^{(r)}_{b,k,\ell,h,2}}\right)\right)\right.\right. \\
& \left.+\frac{\eta^{(r)}_{a,k,\ell,h,1}}{\eta^{(r)}_{a,k,\ell,h,2}}\frac{\eta^{(r)}_{b,k,\ell,h,1}}{\eta^{(r)}_{b,k,\ell,h,2}}\psi'\left(\frac{\eta^{(r)}_{a,k,\ell,h,1}}{\eta^{(r)}_{a,k,\ell,h,2}}+\frac{\eta^{(r)}_{b,k,\ell,h,1}}{\eta^{(r)}_{b,k,\ell,h,2}}\right)\left(\psi\left(\eta^{(r)}_{a,k,\ell,h,1}\right)-\log\left(\eta^{(r)}_{a,k,\ell,h,2}\right)-\log\frac{\eta^{(r)}_{a,k,\ell,h,1}}{\eta^{(r)}_{a,k,\ell,h,2}}\right)\right] \\
& \left.\times\kappa^{-1}\sum_{(i,j)\in\mathcal{T}_{k,\ell}}\eta^{(r)}_{B_{i,j}}\left(1-\eta^{(r)}_{W_{i,j}}\right)\eta^{(r)}_{Z_{i,j}}+c_b\right\} \\
\eta^{(r+1)}_{b,k,\ell,h,2} = & (1-\rho_r)\eta^{(r)}_{b,k,\ell,h,2}+\rho_r\left\{\kappa^{-1}\sum_{(i,j)\in\mathcal{T}_{k,\ell}}\eta_{B^{(r)}_{i,j}}\eta_{W^{(r)}_{i,j}}\eta_{Z^{(r)}_{i,j,h}}\log\left(1-\frac{t_i^{(r)}-t_j^{(r)}}{T_b}\right)+d_b\right\}.
\end{aligned}
\tag{4.20}
$$

We can update $\eta^{(r)}_{a,0,h,1}, \eta^{(r)}_{a,0,h,2}, \eta^{(r)}_{b,0,h,1}, \eta^{(r)}_{b,0,h,2}$ in the same way:

$$\begin{aligned}
\eta^{(r+1)}_{a,0,h,1} = (1-\rho_r)\eta^{(r)}_{a,0,h,1} + \rho_r \Bigg\{ & \left[\frac{\eta^{(r)}_{a,0,h,1}}{\eta^{(r)}_{a,0,h,2}} \left(\psi\left(\frac{\eta^{(r)}_{a,k,\ell,h,1}}{\eta^{(r)}_{a,0,h,2}} + \frac{\eta^{(r)}_{b,0,h,1}}{\eta^{(r)}_{b,0,h,2}} \right) - \psi\left(\frac{\eta^{(r)}_{a,0,h,1}}{\eta^{(r)}_{a,0,h,2}} \right) \right) \right. \\
& \left. + \frac{\eta^{(r)}_{a,0,h,1}}{\eta^{(r)}_{a,0,h,2}} \frac{\eta^{(r)}_{b,0,h,1}}{\eta^{(r)}_{b,0,h,2}} \psi'\left(\frac{\eta^{(r)}_{a,0,h,1}}{\eta^{(r)}_{a,0,h,2}} + \frac{\eta^{(r)}_{b,0,h,1}}{\eta^{(r)}_{b,0,h,2}} \right) \left(\psi\left(\eta^{(r)}_{b,0,h,1} \right) - \log\left(\eta^{(r)}_{b,0,h,2} \right) - \log \frac{\eta^{(r)}_{b,0,h,1}}{\eta^{(r)}_{b,0,h,2}} \right) \right] \\
& \times \kappa^{-1} \sum_{k=1}^{K} \sum_{\ell=1}^{K} \sum_{(i,j)\in\mathcal{T}_{k,\ell}} \eta_{B^{(r)}_{i,j}} \eta_{W^{(r)}_{i,j}} \eta_{Z^{(r)}_{i,j,h}} + c_a \Bigg\}
\end{aligned}$$

$$\eta^{(r+1)}_{a,0,h,2} = (1-\rho_r)\eta^{(r)}_{a,0,h,2} + \rho_r \left\{ \kappa^{-1} \sum_{k=1}^{K} \sum_{\ell=1}^{K} \sum_{(i,j)\in\mathcal{T}_{k,\ell}} \eta_{B^{(r)}_{i,j}} \left(1 - \eta_{W^{(r)}_{i,j}} \right) \eta_{Z^{(r)}_{i,j,h}} \log\left(\frac{t^{(r)}_i - t^{(r)}_j}{T_b} \right) + d_a \right\}. \tag{4.21}$$

$$\begin{aligned}
\eta^{(r+1)}_{b,0,h,1} = (1-\rho_r)\eta^{(r)}_{b,0,h,1} + \rho_r \Bigg\{ & \left[\frac{\eta^{(r)}_{b,0,h,1}}{\eta^{(r)}_{b,0,h,2}} \left(\psi\left(\frac{\eta^{(r)}_{a,k,\ell,h,1}}{\eta^{(r)}_{a,0,h,2}} + \frac{\eta^{(r)}_{b,0,h,1}}{\eta^{(r)}_{b,0,h,2}} \right) - \psi\left(\frac{\eta^{(r)}_{b,0,h,1}}{\eta^{(r)}_{b,0,h,2}} \right) \right) \right. \\
& \left. + \frac{\eta^{(r)}_{a,0,h,1}}{\eta^{(r)}_{a,0,h,2}} \frac{\eta^{(r)}_{b,0,h,1}}{\eta^{(r)}_{b,0,h,2}} \psi'\left(\frac{\eta^{(r)}_{a,0,h,1}}{\eta^{(r)}_{a,0,h,2}} + \frac{\eta^{(r)}_{b,0,h,1}}{\eta^{(r)}_{b,0,h,2}} \right) \left(\psi\left(\eta^{(r)}_{a,0,h,1} \right) - \log\left(\eta^{(r)}_{a,0,h,2} \right) - \log \frac{\eta^{(r)}_{a,0,h,1}}{\eta^{(r)}_{a,0,h,2}} \right) \right] \\
& \times \kappa^{-1} \sum_{k=1}^{K} \sum_{\ell=1}^{K} \sum_{(i,j)\in\mathcal{T}_{k,\ell}} \eta_{B^{(r)}_{i,j}} \eta_{W^{(r)}_{i,j}} \eta_{Z^{(r)}_{i,j,h}} + c_b \Bigg\}
\end{aligned}$$

$$\eta^{(r+1)}_{b,0,h,2} = (1-\rho_r)\eta^{(r)}_{b,0,h,2} + \rho_r \left\{ \kappa^{-1} \sum_{k=1}^{K} \sum_{\ell=1}^{K} \sum_{(i,j)\in\mathcal{T}_{k,\ell}} \eta_{B^{(r)}_{i,j}} \left(1 - \eta_{W^{(r)}_{i,j}} \right) \eta_{Z^{(r)}_{i,j,h}} \log\left(\frac{t^{(r)}_i - t^{(r)}_j}{T_b} \right) + d_b \right\}. \tag{4.22}$$

Updating $\boldsymbol{\eta}_W, \boldsymbol{\eta}_Z$

Let $Q(t \mid \eta_{a,1}, \eta_{a,2}, \eta_{b,1}, \eta_{b,2}) := \mathrm{E}_q \left[f_{\mathrm{Beta}} \left(t \mid a, b; T_0 \right) \right]$. Based on the approximation in Section 4.2, we have

$$
\begin{aligned}
Q(t\,|\eta_{a,1},\eta_{a,2},\eta_{b,1},\eta_{b,2}) \approx & -\log \mathrm{Be}\left(\frac{\eta_{a,1}}{\eta_{a,2}},\frac{\eta_{b,1}}{\eta_{b,2}}\right)\Gamma\left(\frac{\eta_{a,1}}{\eta_{a,2}}\right)\Gamma\left(\frac{\eta_{b,1}}{\eta_{b,2}}\right) \\
& + \frac{\eta_{a,1}}{\eta_{a,2}}\left[\psi\left(\frac{\eta_{a,1}}{\eta_{a,2}}+\frac{\eta_{b,1}}{\eta_{b,2}}\right)-\psi\left(\frac{\eta_{a,1}}{\eta_{a,2}}\right)\right]\left(\psi\left(\eta_{a,1}\right)-\log\left(\eta_{a,2}\right)-\log\frac{\eta_{a,1}}{\eta_{a,2}}\right) \\
& + \frac{\eta_{b,1}}{\eta_{b,2}}\left[\psi\left(\frac{\eta_{a,1}}{\eta_{a,2}}+\frac{\eta_{b,1}}{\eta_{b,2}}\right)-\psi\left(\frac{\eta_{b,1}}{\eta_{b,2}}\right)\right]\left(\psi\left(\eta_{b,1}\right)-\log\left(\eta_{b,2}\right)-\log\frac{\eta_{b,1}}{\eta_{b,2}}\right) \\
& + \frac{1}{2}\frac{\eta_{a,1}}{\eta_{a,2}}\left[\psi'\left(\frac{\eta_{a,1}}{\eta_{a,2}}+\frac{\eta_{b,1}}{\eta_{b,2}}\right)-\psi'\left(\frac{\eta_{a,1}}{\eta_{a,2}}\right)\right]\left[\left(\psi\left(\eta_{a,1}\right)-\log\left(\eta_{a,1}\right)\right)^2+\psi'\left(\eta_{a,1}\right)\right] \\
& + \frac{1}{2}\frac{\eta_{b,1}}{\eta_{b,2}}\left[\psi'\left(\frac{\eta_{a,1}}{\eta_{a,2}}+\frac{\eta_{b,1}}{\eta_{b,2}}\right)-\psi'\left(\frac{\eta_{b,1}}{\eta_{b,2}}\right)\right]\left[\left(\psi\left(\eta_{b,1}\right)-\log\left(\eta_{b,1}\right)\right)^2+\psi'\left(\eta_{b,1}\right)\right] \\
& + \frac{\eta_{a,1}}{\eta_{a,2}}\cdot\frac{\eta_{b,1}}{\eta_{b,2}}\cdot\psi'\left(\frac{\eta_{a,1}}{\eta_{a,2}}+\frac{\eta_{b,1}}{\eta_{b,2}}\right)\left(\psi\left(\eta_{a,1}\right)-\log\left(\eta_{a,2}\right)-\log\frac{\eta_{a,1}}{\eta_{a,2}}\right)\left(\psi\left(\eta_{b,1}\right)-\log\left(\eta_{b,2}\right)-\log\frac{\eta_{b,1}}{\eta_{b,2}}\right) \\
& + \left(\frac{\eta_{a,1}}{\eta_{a,2}}-1\right)\log t+\left(\frac{\eta_{b,1}}{\eta_{b,2}}-1\right)\log\left(T_0-t\right)-\left(\frac{\eta_{a,1}}{\eta_{a,2}}+\frac{\eta_{b,1}}{\eta_{b,2}}-1\right)\log T_0.
\end{aligned}
\tag{4.23}
$$

$$
\eta_{Z_{i,j,h,0}}^{(r+1)} \propto \frac{\psi(\eta_{p,0,h}^{(r+1)})}{\psi(\sum_{h=1}^{L}\eta_{p,0,h}^{(r+1)})} + Q(t_j^{(r)}-t_i^{(r)} \mid \eta_{a_{0,h,1}}^{(r+1)},\eta_{a_{0,h,2}}^{(r+1)},\eta_{b_{0,h,1}}^{(r+1)},\eta_{b_{0,h,2}}^{(r+1)}), \quad h=1,\ldots,L_0.
$$

$$
\eta_{Z_{i,j,h,1}}^{(r+1)} \propto \frac{\psi(\eta_{p,k,\ell,h}^{(r+1)})}{\psi(\sum_{h=1}^{L}\eta_{p,k,\ell,h}^{(r+1)})} + Q(t_j^{(r)}-t_i^{(r)} \mid \eta_{a_{k,\ell,h,1}}^{(r+1)},\eta_{a_{k,\ell,h,2}}^{(r+1)},\eta_{b_{k,\ell,h,1}}^{(r+1)},\eta_{b_{k,\ell,h,2}}^{(r+1)}), \quad h=1,\ldots,L.
$$

Thus, we can update $\boldsymbol{\eta}_W$, $\boldsymbol{\eta}_Z$ using the following formula:

$$
\tilde{\eta}_{W_{i,j,1}}^{(r+1)} = \sum_{h=1}^{L}\left[\frac{\psi(\eta_{\varepsilon,2}^{(r+1)})}{\psi(\eta_{\varepsilon,1}^{(r+1)}+\eta_{\varepsilon,2}^{(r+1)})} + \frac{\psi(\eta_{p,k,\ell,h}^{(r+1)})}{\psi(\sum_{h=1}^{L}\eta_{p,k,\ell,h}^{(r+1)})} + Q(t_j^{(r)}-t_i^{(r)} \mid \eta_{a_{k,\ell,h,1}}^{(r+1)},\eta_{a_{k,\ell,h,2}}^{(r+1)},\eta_{b_{k,\ell,h,1}}^{(r+1)},\eta_{b_{k,\ell,h,2}}^{(r+1)})\right]
$$

$$
\tilde{\eta}_{W_{i,j,0}}^{(r+1)} = \sum_{h=1}^{L_0}\left[\frac{\psi(\eta_{\varepsilon,1}^{(r+1)})}{\psi(\eta_{\varepsilon,1}^{(r+1)}+\eta_{\varepsilon,2}^{(r+1)})} + \frac{\psi(\eta_{p,0,h}^{(r+1)})}{\psi(\sum_{h=1}^{L_0}\eta_{p,0,h}^{(r+1)})} + Q(t_j^{(r)}-t_i^{(r)} \mid \eta_{a_{0,h,1}}^{(r+1)},\eta_{a_{0,h,2}}^{(r+1)},\eta_{b_{0,h,1}}^{(r+1)},\eta_{b_{0,h,2}}^{(r+1)})\right]
$$

$$
\eta_{W_{i,j}}^{(r+1)} = \frac{\tilde{\eta}_{W_{i,j,1}}^{(r+1)}}{\tilde{\eta}_{W_{i,j,0}}^{(r+1)}+\tilde{\eta}_{W_{i,j,1}}^{(r+1)}}
$$

Updating $\boldsymbol{\eta}_B$

For $i=1,\ldots,n, \quad j=1,\ldots,i$, we have:

$$
\begin{aligned}
\tilde{\eta}_{B_{j,i}}^{(r+1)} &= \log\left(\frac{t_j^{(r)} - t_i^{(r)}}{T}\right)\left[\sum_{h=1}^{L}\eta_{Z_{i,j,h,1}}^{(r+1)}\eta_{W_{i,j}}^{(r+1)}\left(\frac{\eta_{a,k,\ell,h,1}^{(r+1)}}{\eta_{a,k,\ell,h,2}^{(r+1)}} - 1\right) + \sum_{h=1}^{L_0}\eta_{Z_{i,j,h,0}}^{(r+1)}(1-\eta_{W_{i,j}}^{(r+1)})\left(\frac{\eta_{a,0,h,1}^{(r+1)}}{\eta_{a,0,h,2}^{(r+1)}} - 1\right)\right] \\
&\quad + \log\left(1 - \frac{t_j^{(r)} - t_i^{(r)}}{T}\right)\left[\sum_{h=1}^{L}\eta_{Z_{i,j,h,1}}^{(r+1)}\eta_{W_{i,j}}^{(r+1)}\left(\frac{\eta_{b,k,\ell,h,1}^{(r+1)}}{\eta_{b,k,\ell,h,2}^{(r+1)}} - 1\right) + \sum_{h=1}^{L_0}\eta_{Z_{i,j,h,0}}^{(r+1)}(1-\eta_{W_{i,j}}^{(r+1)})\left(\frac{\eta_{b,0,h,1}^{(r+1)}}{\eta_{b,0,h,2}^{(r+1)}} - 1\right)\right] \\
&\quad + \sum_{h=1}^{L_0} Q\left(t_j^{(r)} - t_i^{(r)} \mid \eta_{a_{0,h,1}}^{(r+1)}, \eta_{a_{0,h,2}}^{(r+1)}, \eta_{b_{0,h,1}}^{(r+1)}, \eta_{b_{0,h,2}}^{(r+1)}\right)(1-\eta_{W_{i,j}})\eta_{Z_{i,j,h,0}} \\
&\quad + \sum_{h=1}^{L} Q\left(t_j^{(r)} - t_i^{(r)} \mid \eta_{a_{k,\ell,h,1}}^{(r+1)}, \eta_{a_{k,\ell,h,2}}^{(r+1)}, \eta_{b_{k,\ell,h,1}}^{(r+1)}, \eta_{b_{k,\ell,h,2}}^{(r+1)}\right)\eta_{W_{i,j}}\eta_{Z_{i,j,h,1}} \\
&\quad + \psi(\eta_{\alpha,k,\ell,1}) - \log(\eta_{\alpha,k,\ell,2}) - \log T_0 \\
\tilde{\eta}_{B_{j,j}}^{(r+1)} &= \psi(\eta_{\mu,k,1}) - \log(\eta_{\mu,k,2}).
\end{aligned}
$$

Finally, we have $\eta_{B_{j,i}}^{(r+1)} = \frac{\tilde{\eta}_{B_{j,i}}^{(r+1)}}{\sum_{j=1}^{i}\tilde{\eta}_{B_{j,i}}^{(r+1)}}$.

4.4.3 Details of the MCMC algorithm

Sampling $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$

Given the branching structure, the full conditional posteriors for MHP parameters $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$ are conjugate to their Gamma priors:

$$
\begin{aligned}
\mu_\ell \mid \mathbf{B}, \mathbf{X} &\sim \text{Gamma}(e_\ell + |I_\ell|, f_\ell + T), \quad \ell = 1, \ldots, K \\
\alpha_{k,\ell} \mid \mathbf{B}, \mathbf{X} &\sim \text{Gamma}\left(g_{k,\ell} + |O_{k,\ell}|, h_{k,\ell} + \sum_{d_i = k}\tilde{\Phi}_{k,\ell}\left(T - t_i\right)\right), \quad k, \ell = 1, \ldots, K.
\end{aligned}
$$

Sampling $\boldsymbol{b}$

Given the model parameters, we can update the branching structure using the following formula. For $j = 1, \ldots, n, i = 1, \ldots j$, we have:

$$
\begin{aligned}
p\left(\mathbf{B}_{j,j} = 1, \mathbf{B}_{j,-j} = 0 \mid \boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{p}, \mathbf{X}\right) &= \frac{\mu_{d_j}}{\mu_{d_j} + \sum_{i=1}^{j-1}\alpha_{d_i,d_j}\tilde{\phi}_{d_i,d_j}(t_j - t_i)} \\
p\left(\mathbf{B}_{j,i} = 1, \mathbf{B}_{j,-i} = 0 \mid \boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{p}, \mathbf{X}\right) &= \frac{\alpha_{d_i,d_j}\tilde{\phi}_{d_i,d_j}(t_j - t_i)}{\mu_{d_j} + \sum_{i \neq j}\alpha_{d_i,d_j}\tilde{\phi}_{d_i,d_j}(t_j - t_i)}.
\end{aligned}
$$

Sampling $\mathbf{Z}$

Given $\boldsymbol{b}$ and model parameters $\boldsymbol{a}^0, \boldsymbol{a}^{k,\ell}, \boldsymbol{b}^0, \boldsymbol{b}^{k,\ell}$, $\boldsymbol{p}^0$, $\boldsymbol{p}^{k,\ell}$ and ε, we can generate posterior samples of the latent allocation variables sequentially, from the following discrete distribution. If $B_{i,j} = 1$ and $(i,j) \in \mathcal{T}_{i,j}$,

$$P(W_{i,j} = 0 \mid \mathbf{B}, \mathbf{X}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{p}, \varepsilon)$$
$$= \frac{\sum_{h=1}^{L_0} \varepsilon p_h^0 f_{\text{ Beta }}\left(t_j - t_i \mid a_h^0, b_h^0\right)}{\sum_{h=1}^{L_0} \varepsilon p_h^0 f_{\text{ Beta }}\left(t_j - t_i \mid a_h^0, b_h^0\right) + \sum_{h=1}^{L} (1-\varepsilon) p_h^{k,\ell} f_{\text{ Beta }}\left(t_j - t_i \mid a_h^{k,\ell}, b_h^{k,\ell}\right)}$$

$$P(W_{i,j} = 1 \mid \mathbf{B}, \mathbf{X}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{p}, \varepsilon)$$
$$= \frac{\sum_{h=1}^{L} (1-\varepsilon) p_h^{k,\ell} f_{\text{ Beta }}\left(t_j - t_i \mid a_h^{k,\ell}, b_h^{k,\ell}\right)}{\sum_{h=1}^{L_0} \varepsilon p_h^0 f_{\text{ Beta }}\left(t_j - t_i \mid a_h^0, b_h^0\right) + \sum_{h=1}^{L} (1-\varepsilon) p_h^{k,\ell} f_{\text{ Beta }}\left(t_j - t_i \mid a_h^{k,\ell}, b_h^{k,\ell}\right)}$$

$$P(Z_{i,j} = h \mid W_{i,j} = 0, \mathbf{B}, \mathbf{X}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{p}, \varepsilon) = \frac{p_h^0 f_{\text{ Beta }}\left(t_j - t_i \mid a_h^0, b_h^0\right)}{\sum_{h=1}^{L_0} p_h^0 f_{\text{ Beta }}\left(t_j - t_i \mid a_h^0, b_h^0\right)}, \text{ for } h = 1, \ldots, L_0$$

$$P(Z_{i,j} = h \mid W_{i,j} = 1, \mathbf{B}, \mathbf{X}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{p}, \varepsilon) = \frac{p_h^{k,\ell} f_{\text{ Beta }}\left(t_j - t_i \mid a_h^{k,\ell}, b_h^{k,\ell}\right)}{\sum_{h=1}^{L} p_h^{k,\ell} f_{\text{ Beta }}\left(t_j - t_i \mid a_h^{k,\ell}, b_h^{k,\ell}\right)}, \text{ for } h = 1, \ldots, L.$$

Sampling $\boldsymbol{a}$ *and* $\boldsymbol{b}$

Up to a normalizing constant and approximation of the compensator, the full conditional distribution for $\boldsymbol{a}^0, \boldsymbol{a}^{k,\ell}, \boldsymbol{b}^0$ and $\boldsymbol{b}^{k,\ell}$ can be written as

$$
\begin{aligned}
p(a_h^{k,\ell} \mid \mathbf{X}, \mathbf{B}, \mathbf{Z}, \cdot) \propto & \left[\frac{\Gamma\left(a_h^{k,\ell} + b_h^{k,\ell}\right)}{\Gamma\left(a_h^{k,\ell}\right)} \right]^{N_h^{k,\ell}} \\
& \times \prod_{(i,j)\in\mathcal{T}_{k,\ell}} \left[(t_j - t_i)^{a_h^{k,\ell}-1} \right]^{B_{i,j}W_{i,j}I(Z_{i,j}=h)} \left(a_h^{k,\ell}\right)^{c_a-1} e^{-d_a a_h^{k,\ell}} \\
p(a_h^0 \mid \mathbf{X}, \mathbf{B}, \mathbf{Z}, \cdot) \propto & \left[\frac{\Gamma\left(a_h^0 + b_h^0\right)}{\Gamma\left(a_h^0\right)} \right]^{N_h^0} \\
& \times \prod_{k=1}^{K} \prod_{\ell=1}^{K} \prod_{(i,j)\in\mathcal{T}_{k,\ell}} \left[(t_j - t_i)^{a_h^0-1} \right]^{B_{i,j}(1-W_{i,j})I(Z_{i,j}=h)} \left(a_h^0\right)^{c_a-1} e^{-d_a a_h^0} \\
p(b_h^{k,\ell} \mid \mathbf{X}, \mathbf{B}, \mathbf{Z}, \cdot) \propto & \left[\frac{\Gamma\left(a_h^{k,\ell} + b_h^{k,\ell}\right)}{\Gamma\left(b_h^{k,\ell}\right)} \right]^{N_h^{k,\ell}} \\
& \times \prod_{(i,j)\in\mathcal{T}_{k,\ell}} \left[[T_b - (t_j - t_i)]^{b_h^{k,\ell}-1} \right]^{B_{i,j}W_{i,j}I(Z_{i,j}=h)} \left(b_h^{k,\ell}\right)^{c_b-1} e^{-d_b b_h^{k,\ell}} \\
p(b_h^0 \mid \mathbf{X}, \mathbf{B}, \mathbf{Z}, \cdot) \propto & \left[\frac{\Gamma\left(a_h^0 + b_h^0\right)}{\Gamma\left(b_h^0\right)} \right]^{N_h^0} \\
& \times \prod_{k=1}^{K} \prod_{\ell=1}^{K} \prod_{(i,j)\in\mathcal{T}_{k,\ell}} \left[[T_b - (t_j - t_i)]^{b_h^0-1} \right]^{B_{i,j}(1-W_{i,j})I(Z_{i,j}=h)} \left(b_h^0\right)^{c_b-1} e^{-d_b b_h^0},
\end{aligned}
$$

where

$$
N_h^{k,\ell} = \sum_{(i,j)\in\mathcal{T}_{k,\ell}} B_{i,j}(1 - W_{i,j}) I\left(Z_{i,j} = h\right), N_h^0 = \sum_{k=1}^{K}\sum_{\ell=1}^{K}\sum_{(i,j)\in\mathcal{T}_{k,\ell}} B_{i,j}W_{i,j} I\left(Z_{i,j} = h\right).
$$

As the posterior distributions for $\boldsymbol{a}^{k,\ell}, \boldsymbol{a}^0, \boldsymbol{b}^{k,\ell}, \boldsymbol{b}^0$ are not conjugate to their priors, we update these parameters through Metropolis-within-Gibbs updates on the log scale.

Sampling $\boldsymbol{p}$ *and* $\boldsymbol{\varepsilon}$

Finally, the full conditional posteriors for $\boldsymbol{p}$ and ε are conjugate to their Dirichlet and Beta priors:

$$\begin{aligned}
\boldsymbol{p}^0 \mid \mathbf{B}, \mathbf{W}, \mathbf{Z} &\sim \text{Dirichlet}\left(\frac{\alpha_{DP}}{L_0} + N_1^0, \ldots, \frac{\alpha_{DP}}{L_0} + N_{L_0}^0\right) \\
\boldsymbol{p}^{k,\ell} \mid \mathbf{B}, \mathbf{W}, \mathbf{Z} &\sim \text{Dirichlet}\left(\frac{\alpha_{DP}}{L} + N_1^{k,\ell}, \ldots, \frac{\alpha_{DP}}{L} + N_L^{k,\ell}\right) \\
\varepsilon \mid \mathbf{B}, \mathbf{W}, \mathbf{Z} &\sim \text{Beta}\left(1 + \sum_{h=1}^{L_0} N_h^0, 1 + \sum_{k=1}^{K}\sum_{\ell=1}^{K}\sum_{h=1}^{L} N_h^{k,\ell}\right).
\end{aligned}$$

4.5 Simulation Studies

Data generating mechanisms. The data is generated from a multivariate Hawkes process with $K = 2$ dimensions. All scenarios we consider share the following parameter values: $\boldsymbol{\alpha} = \begin{bmatrix} 0.6 & 0.15 \\ 0.3 & 0.6 \end{bmatrix}$, $\boldsymbol{\mu} = \begin{bmatrix} 0.05 \\ 0.1 \end{bmatrix}$ and $T = 15000$. For the triggering kernels, we consider two scenarios based on whether the Beta mixture component is correctly specified. First, we consider scenarios where the excitation functions kernels are correctly-specified as mixtures of Beta distributions:

$$\tilde{\phi}_{k,\ell}^{\text{true}}(t) = \varepsilon_{\text{true}}\text{Beta}(t \mid a_{\text{true}}^0, b_{\text{true}}^0, T_0) + (1 - \varepsilon_{\text{true}})\text{Beta}(t \mid a_{\text{true}}^{k,\ell}, b_{\text{true}}^{k,\ell}, T_0)$$

where

$$a_{\text{true}}^0 = 1, \quad b_{\text{true}}^0 = 4, \quad \begin{bmatrix} a_{\text{true}}^{11} & a_{\text{true}}^{12} \\ a_{\text{true}}^{21} & a_{\text{true}}^{22} \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 1.5 & 1 \end{bmatrix}, \quad \begin{bmatrix} b_{\text{true}}^{11} & b_{\text{true}}^{12} \\ b_{\text{true}}^{21} & b_{\text{true}}^{22} \end{bmatrix} = \begin{bmatrix} 6 & 1 \\ 5 & 1 \end{bmatrix}, \quad T_0 = 1.$$

We also consider a misspecified model with exponential excitation functions :

$$\tilde{\phi}_{k,\ell}^{\text{true}}(t) = \varepsilon_{\text{true}} \exp(-t) + (1 - \varepsilon_{\text{true}}) \exp(-\lambda_{\text{true}}^{j,k} t),$$

where $\begin{bmatrix} \lambda_{\text{true}}^{11} & \lambda_{\text{true}}^{12} \\ \lambda_{\text{true}}^{21} & \lambda_{\text{true}}^{22} \end{bmatrix} = \begin{bmatrix} 2 & 0.8 \\ 0.8 & 2 \end{bmatrix}$. Additionally, for each of these two scenarios, we consider five values of $\varepsilon_{\text{true}}$, $\{0, 0.2, 0.5, 0.8, 1\}$.

Benchmark methods. We fit our model using the two computational approaches discussed in Section 4.4. We also compare our method where ε is treated as an unknown parameter ('RANDOM'), to two benchmark versions, one where there is no information borrowing ('IDIO', which corresponds to fixed $\varepsilon = 0$) or the triggering kernels are identical ('COMMON', which corresponds to fixed $\varepsilon = 1$). Each of IDIO and COMMON are in turn fitted using versions of both the MCMC and the SGVI algorithms discussed in Section 4.4. Finally, we consider a frequentist benchmark method based on piecewise basis kernels using the EM algorithm (EM-BK, see 170).

Performance metrics. We use the root mean integrated squared error (RMISE) as a metric for point estimation accuracy:

$$\text{RMISE}(\boldsymbol{\phi}) = \frac{1}{K^2} \sum_{k=1}^{K} \sum_{\ell=1}^{K} \sqrt{\int_0^{+\infty} \left(\phi_{k,\ell}^{\text{true}}(x) - \hat{\phi}_{k,\ell}(x) \right)^2 \, \mathrm{d}x}.$$

This is just the L_2 distance between the excitation functions and its estimate (which, for Bayesian procedures, corresponds to the posterior mean), averaged over all dimension pairs. In practice, we approximate the integral involved in the definition of $\text{RMISE}(\boldsymbol{\phi})$ by averaging the value of the function over a fine grid.

On the other hand, we use the average coverage rate (ACR) and the interval score (IS) to evaluate uncertainty estimation. ACR is defined as the proportion of correct coverages of the triggering kernels evaluated on the grid points, averaged over all the dimension pairs. IS [59] is a generalization of ACR that penalizes wider interval lengths and low coverage rates.

Results. Tables 4.1 through 4.3 show the RMISE, coverage rates and interval scores across five different values of $\varepsilon_{\text{true}}$ under the correctly-specified data generating mechanism for the various approaches under consideration. Similarly, Tables 4.4 through 4.6 show the results under the mis-specified data generating mechanism. For the MCMC algorithm under the correctly-specified scenario, 'IDIO' dominates the other two methods when ($\varepsilon_{\text{true}} = 0$), as it has the lowest RMISE. Likewise, 'COMMON' dominates under the scenario where ($\varepsilon_{\text{true}} = 1$). This is not surprising, as in both cases the best performing model matches the

true model. However, for scenarios where the true data is a mixture of a common and an idiosyncratic components, our method ('RANDOM') shows the lowest RMISE. The ranking of the models under stochastic variational inference is similar, but we see slightly higher RIMSE values across the board. Finally, RMISE values for EM-BK are much higher than those for the MCMC algorithm in all scenarios. The 'IDIO' method for MCMC is closer to the nominal coverages in most scenarios, but have slightly lower interval scores due to having wider intervals. The SVI algorithm tends to underestimate parameter uncertainty, causing much lower coverages and higher interval scores. The results for the mis-specified case are similar.

		MCMC			SVI		EM-BK
$\varepsilon_{\text{true}}$	RANDOM	IDIO	COMMON	RANDOM	IDIO	COMMON	-
0	0.097	0.096	0.775	0.279	0.182	0.859	0.828
	(0.016)	(0.021)	(0.001)	(0.032)	(0.129)	(0.011)	(0.297)
0.2	0.102	0.102	0.621	0.225	0.243	0.723	0.505
	(0.014)	(0.012)	(0.002)	(0.034)	(0.082)	(0.020)	(0.207)
0.5	0.102	0.109	0.391	0.216	0.241	0.554	0.200
	(0.010)	(0.010)	(0.001)	(0.042)	(0.081)	(0.012)	(0.076)
0.8	0.080	0.101	0.163	0.296	0.329	0.399	0.126
	(0.007)	(0.009)	(0.002)	(0.012)	(0.139)	(0.144)	(0.014)
1	0.051	0.010	0.047	0.433	0.406	0.342	0.139
	(0.012)	(0.010)	(0.011)	(0.009)	(0.143)	(0.219)	(0.022)

Table 4.1: RMISE as a point estimation accuracy metric for all methods under five true information-borrowing ratios. The values in the grid cells are the average over 10 independently generated datasets, and the standard deviation is shown in the brackets.

	MCMC			SVI		
$\varepsilon_{\text{true}}$	RANDOM	IDIO	COMMON	RANDOM	IDIO	COMMON
0	0.728	0.776	0.035	0.138	0.351	0.013
	(0.096)	(0.089)	(0.003)	(0.106)	(0.189)	(0.003)
0.2	0.796	0.843	0.045	0.059	0.263	0.018
	(0.089)	(0.116)	(0.005)	(0.017)	(0.084)	(0.004)
0.5	0.822	0.902	0.071	0.048	0.244	0.029
	(0.096)	(0.079)	(0.008)	(0.022)	(0.101)	(0.009)
0.8	0.829	0.870	0.176	0.052	0.244	0.036
	(0.049)	(0.046)	(0.037)	(0.009)	(0.148)	(0.017)
1	0.801	0.896	0.801	0.450	0.193	0.190
	(0.104)	(0.100)	(0.150)	(0.319)	(0.171)	(0.347)

Table 4.2: Coverage rate as an uncertainty estimation accuracy metric for all methods under five true information-borrowing ratios. The values in the grid cells are the average over 10 independently generated datasets, and the standard deviation is shown in the brackets.

	MCMC			SVI		
$\varepsilon_{\text{true}}$	RANDOM	IDIO	COMMON	RANDOM	IDIO	COMMON
0	0.014	0.014	0.629	0.255	0.118	0.719
	(0.005)	(0.008)	(0.003)	(0.002)	(0.098)	(0.005)
0.2	0.014	0.015	0.492	0.293	0.132	0.600
	(0.005)	(0.005)	(0.005)	(0.030)	(0.066)	(0.015)
0.5	0.014	0.014	0.295	0.388	0.132	0.448
	(0.005)	(0.004)	(0.005)	(0.044)	(0.050)	(0.011)
0.8	0.011	0.012	0.100	0.150	0.188	0.331
	(0.002)	(0.002)	(0.002)	(0.009)	(0.088)	(0.128)
1	0.006	0.011	0.005	0.008	0.218	0.276
	(0.003)	(0.004)	(0.002)	(0.005)	(0.067)	(0.188)

Table 4.3: Interval score as an uncertainty estimation accuracy metric for all methods under five true information-borrowing ratios. The values in the grid cells are the average over 10 independently generated datasets, and the standard deviation is shown in the brackets.

	MCMC			SVI			EM-BK
$\varepsilon_{\text{true}}$	RANDOM	IDIO	COMMON	RANDOM	IDIO	COMMON	-
0	0.060	0.063	0.254	0.159	0.170	0.289	0.177
	(0.010)	(0.009)	(0.001)	(0.029)	(0.018)	(0.010)	(0.018)
0.2	0.059	0.062	0.204	0.152	0.164	0.221	0.165
	(0.004)	(0.005)	(0.001)	(0.018)	(0.020)	(0.028)	(0.009)
0.5	0.055	0.061	0.130	0.152	0.160	0.162	0.146
	(0.013)	(0.007)	(0.002)	(0.017)	(0.022)	(0.038)	(0.004)
0.8	0.044	0.063	0.058	0.165	0.128	0.103	0.150
	(0.004)	(0.007)	(0.002)	(0.072)	(0.026)	(0.056)	(0.009)
1	0.027	0.061	0.023	0.087	0.105	0.097	0.153
	(0.004)	(0.009)	(0.004)	(0.060)	(0.012)	(0.064)	(0.007)

Table 4.4: RMISE as a point estimation accuracy metric for all methods under five true information-borrowing ratios for the mis-specified scenario. The values in the grid cells are the average over 10 independently generated datasets, and the standard deviation is shown in the brackets.

	MCMC			SVI		
$\varepsilon_{\text{true}}$	RANDOM	IDIO	COMMON	RANDOM	IDIO	COMMON
0	0.781	0.792	0.044	0.091	0.108	0.064
	(0.053)	(0.090)	(0.020)	(0.040)	(0.040)	(0.030)
0.2	0.813	0.827	0.072	0.068	0.089	0.024
	(0.071)	(0.078)	(0.047)	(0.014)	(0.045)	(0.006)
0.5	0.773	0.840	0.200	0.047	0.080	0.027
	(0.089)	(0.069)	(0.064)	(0.018)	(0.035)	(0.005)
0.8	0.812	0.869	0.493	0.050	0.095	0.044
	(0.06)	(0.072)	(0.057)	(0.021)	(0.020)	(0.024)
1	0.840	0.860	0.759	0.052	0.119	0.052
	(0.098)	(0.085)	(0.166)	(0.017)	(0.017)	(0.014)

Table 4.5: Coverage rate as an uncertainty estimation accuracy metric for all methods under five true information-borrowing ratios. The values in the grid cells are the average over 10 independently generated datasets, and the standard deviation is shown in the brackets.

	MCMC			SVI		
$\varepsilon_{\text{true}}$	RANDOM	IDIO	COMMON	RANDOM	IDIO	COMMON
0	0.003 (0.001)	0.003 (0.001)	0.057 (0.001)	0.046 (0.008)	0.047 (0.006)	0.071 (0.001)
0.2	0.003 (<0.001)	0.003 (<0.001)	0.044 (0.001)	0.047 (0.006)	0.047 (0.008)	0.058 (0.009)
0.5	0.004 (0.002)	0.003 (0.001)	0.026 (<0.001)	0.05 (0.008)	0.047 (0.008)	0.047 (0.016)
0.8	0.003 (0.001)	0.003 (0.001)	0.008 (0.001)	0.053 (0.025)	0.038 (0.011)	0.034 (0.023)
1	0.001 (<0.001)	0.003 (0.001)	0.001 (<0.001)	0.03 (0.025)	0.03 (0.003)	0.035 (0.026)

Table 4.6: Interval score as an uncertainty estimation accuracy metric for all methods under five true information-borrowing ratios. The values in the grid cells are the average over 10 independently generated datasets, and the standard deviation is shown in the brackets.

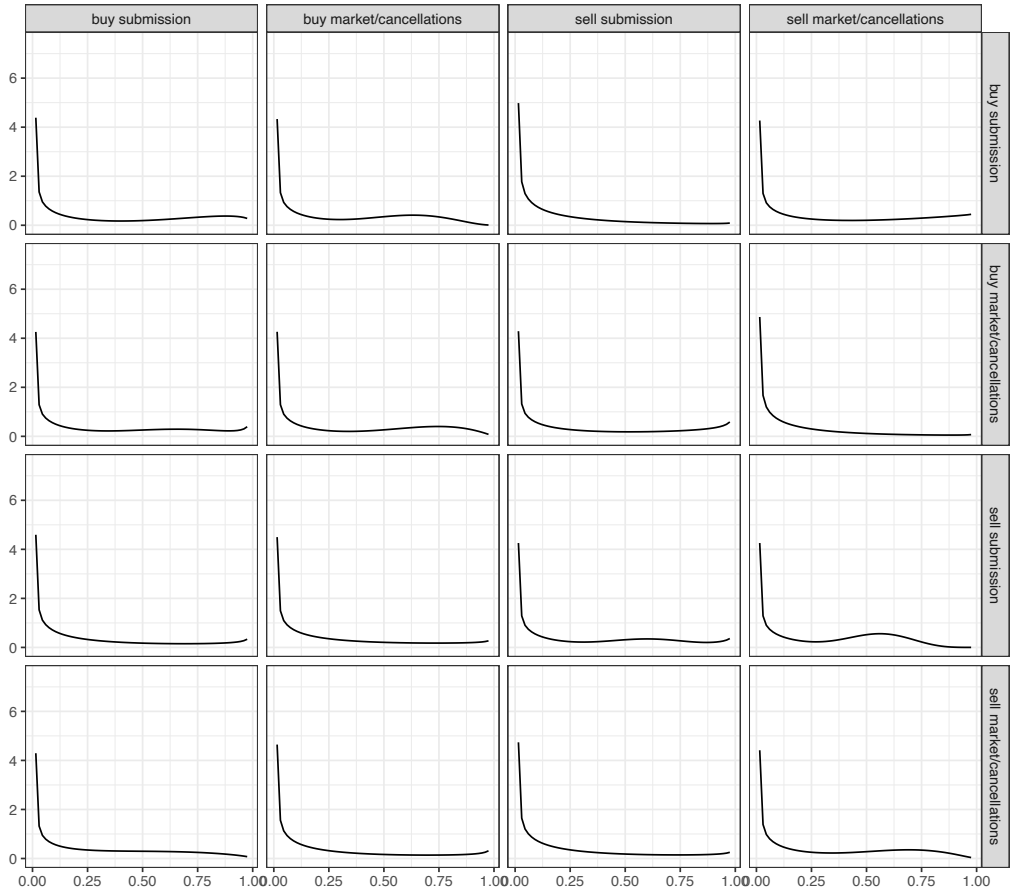

Figure 4.1: Estimated density plots from the SVI algorithm of the cross-dimensional inter-arrival times (within one second) among the four dimensions.

4.6 Application: Modeling order flow in financial markets

In this Section, we apply our method to study the order flow in Amazon's limit order book (LOB). We start by giving a brief introduction of the LOB data structure (we refer the readers to [63] for a detailed description of LOB data). LOB records the placement of limit orders from both buyers and sellers, with the submission time, proposed price and volume. Once a limit order is placed, the order-matching algorithm of the platform attempts to match it with a pre-existing order of the other trade direction. A successful matching is called *market order*. If the placed order does not match with any orders, it remains an

active order and is recorded by LOB until it is matched or *cancelled*. Within the scope of Chapter 4, we will look at the arrival of three event types: placement of active limit orders (submissions), market orders, and order cancellations.

Our LOB data comes from LOBSTER (Limit Order Book System - The Efficient Reconstructor, [73]), which is based on the official NASDAQ Historical TotalView-ITCH sample. The dataset includes the order book for events between 9:30:00 AM and 16:00:00 PM on June 21, 2012. We focus on modeling the order flow of level 1 data, that is, the order with the best $bid(ask)$ prices, defined as the highest (lowest) price at which there is an active limit order. We exclude orders with trading volume lower than 100. The dataset includes 30411 order events with the timestamps (with nanosecond decimal precision). We group the events into four dimensions: buy submissions, buy market orders/cancellations, sell submissions, and sell market orders/cancellations, each containing 8409, 6791, 8801 and 6410 events, respectively. The flow of order events can be seen as a four-dimensional counting process, which can be modelled using an MHP.

To motivate the use of nonparametric estimators for the excitation functions of MHPs in this context, we present in Figure 4.4 the density plots of the cross-dimensional interarrival times among all four dimensions. Note that we excluded the interarrival times over 1 second as events are very unlikely to trigger events in the far future. There are two major implications of the figure. First, the density plots for interarrival times demonstrate various degrees of heterogeneity across dimensions, while having a similar shape. Another discovery is that the majority of the plots are multimodal, with a small 'bump' roughly around the 0.5 second mark; one hypothesis is that this bump is the result of automated algorithms responding to market events.

We fit our model to the dataset using both MCMC and SVI methods. For both methods, we let $T_0 = 1$. We collected 10,000 posterior samples from the MCMC algorithms, with the first 5,000 discarded as burn-in. To prevent convergence to sub-optimal modes, we fit the model to the same dataset under 50 different starting values with different random seeds, and keep the instance with the highest posterior mean marginal likelihood. Figure 4.5 shows the posterior mean for the pointwise triggering kernels. The graphs show different decay rates across dimensions. Remarkably, there are two small bumps for transitions:

from 'buy submissions' to 'buy market order/cancellations' and from 'sell submissions' to 'sell market order/cancellations'. Similarly, we run the SVI algorithm to fit the dataset with 100 different instances and choose the one with the highest ELBO to evaluate the parameter estimation results. We generated 5,000 samples from the variational distribution obtained from the converged model. Figure 4.1 shows the variational mean for the pointwise triggering kernels. Similar to Figure 4.5, the triggering kernels decay differently, and there is a clear bump from 'sell submissions' to 'sell market order/cancellations'.

We also compared the estimation results for the $\boldsymbol{\alpha}$ matrix across both methods. Figure 4.2 shows the heatmap for the point estimates (upper panel), 95% credible interval lengths (lower panel) and Figure 4.3 shows histograms of the spectral radius of the $\boldsymbol{\alpha}$ matrix for the MCMC (left panel) and SVI (right panel) algorithms. The spectral radius $\rho(\boldsymbol{\alpha})$ is defined as the absolute value of the largest eigenvalue of the $\boldsymbol{\alpha}$ matrix: $\rho(\boldsymbol{\alpha}) = \max_{\lambda \in \mathcal{E}(\boldsymbol{\alpha})} |\lambda|$, where $\lambda \in \mathcal{E}(\boldsymbol{\alpha})$ denotes the set of eigenvalues of $\boldsymbol{\alpha}$. One sufficient condition for the stationarity of the MHP is that $\rho(\boldsymbol{\alpha}) < 1$ [67, 1]. Hence, from Figure 4.3, we see that both MCMC and SVI suggest the process we study here is stationary. However, the posterior distribution for the spectral radius under SVI suggest a higher value, and somewhat surprisingly, also a higher variability. On the other hand, Figure 4.2 suggests that there are stronger interactions within the 'buy' or 'sell' transactions. This result is in line with the empirical results in Figure 4.4, where the same structure can be seen.

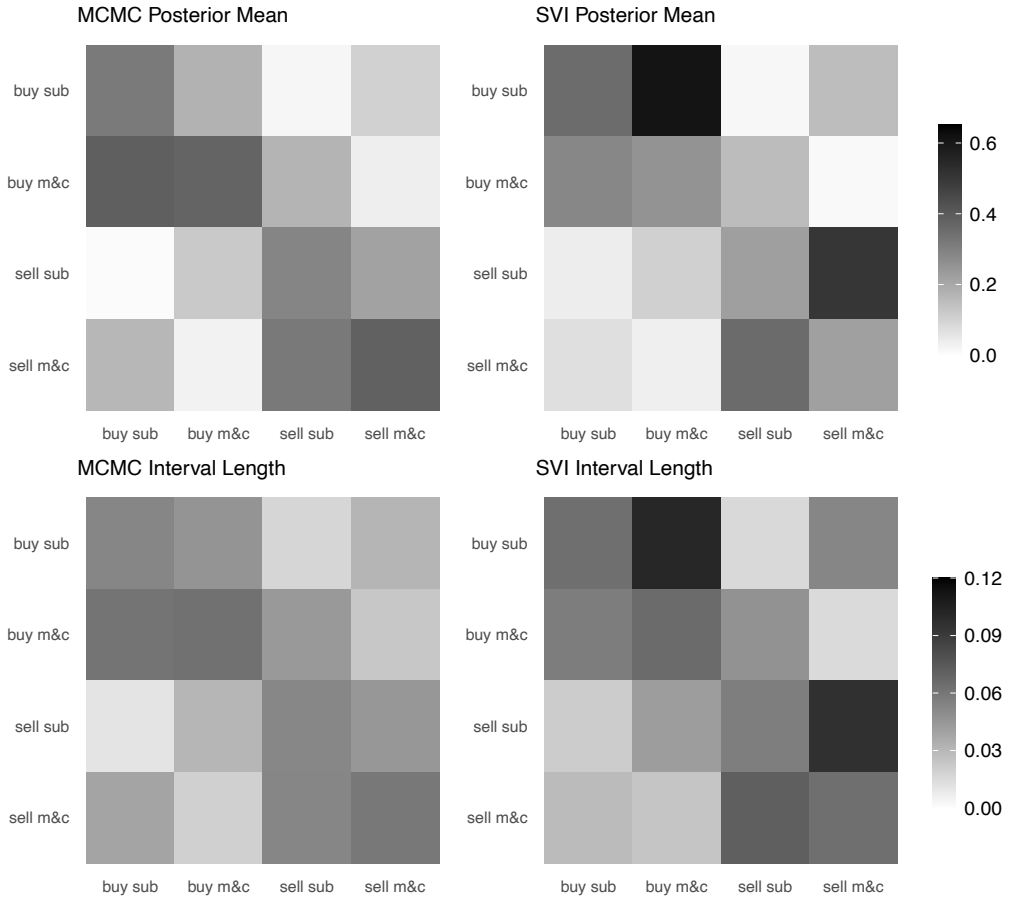

Figure 4.2: Estimation results for the $\boldsymbol{\alpha}$ matrix. The heatmaps show the posterior mean (first row) and average 95% posterior interval lengths (second row) for the values in $\boldsymbol{\alpha}$ by MCMC (left panel) and SVI (right panel) algorithms, for buy submissions ('buy sub'), buy market order/cancellations ('buy m&c'), sell submissions ('sell sub'), sell market order/cancellations ('sell m&c'). y-axis shows the parent events, while x-axis shows the child events. Darker colors corresponds to higher values.

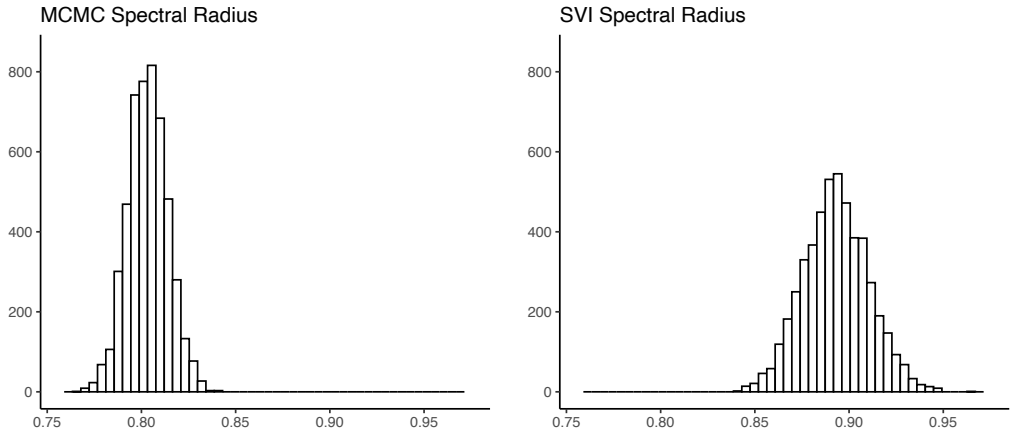

Figure 4.3: Histograms for the spectral radius of the $\boldsymbol{\alpha}$ matrix by the MCMC (left) and SVI (right) algorithm.

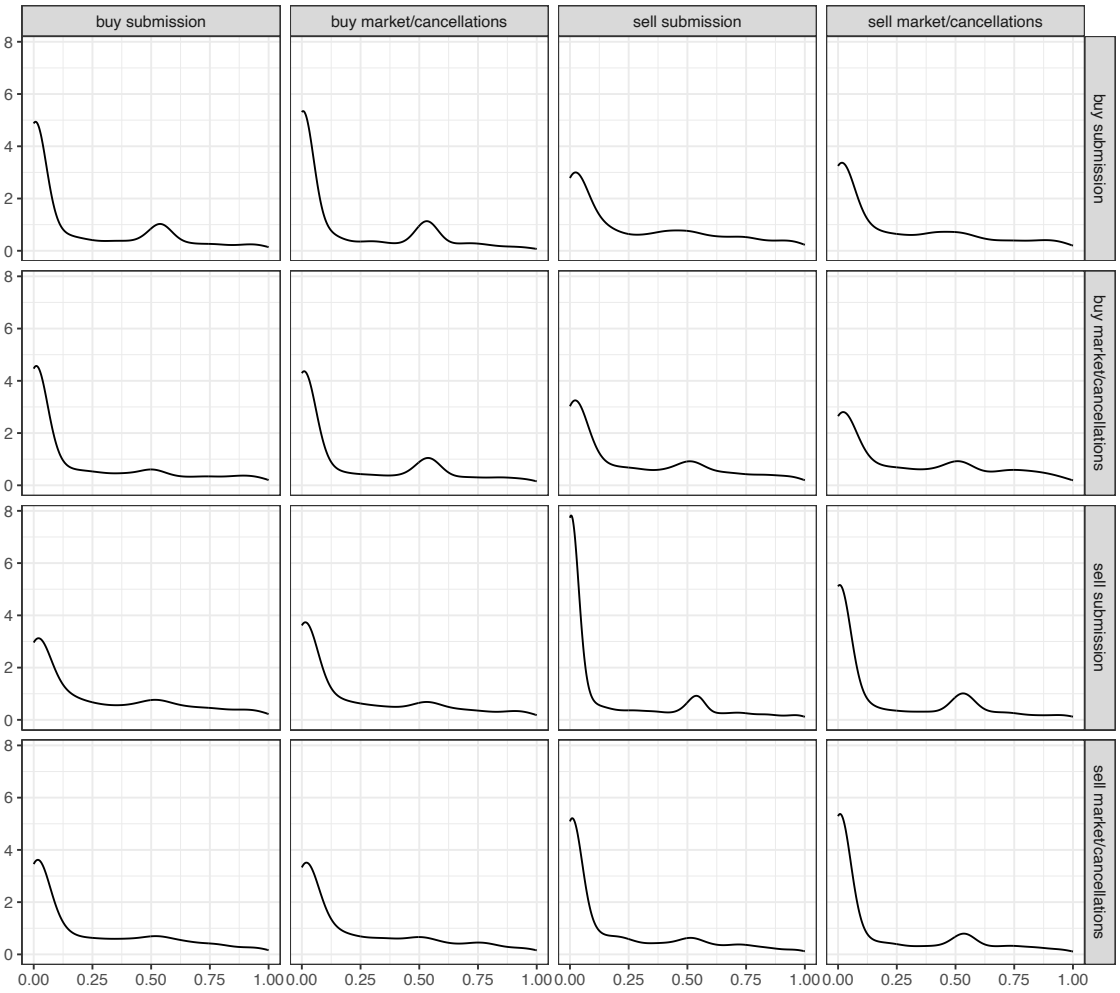

Figure 4.4: Empirical density plots of the cross-dimensional inter-arrival times (within one second) among the four dimensions.

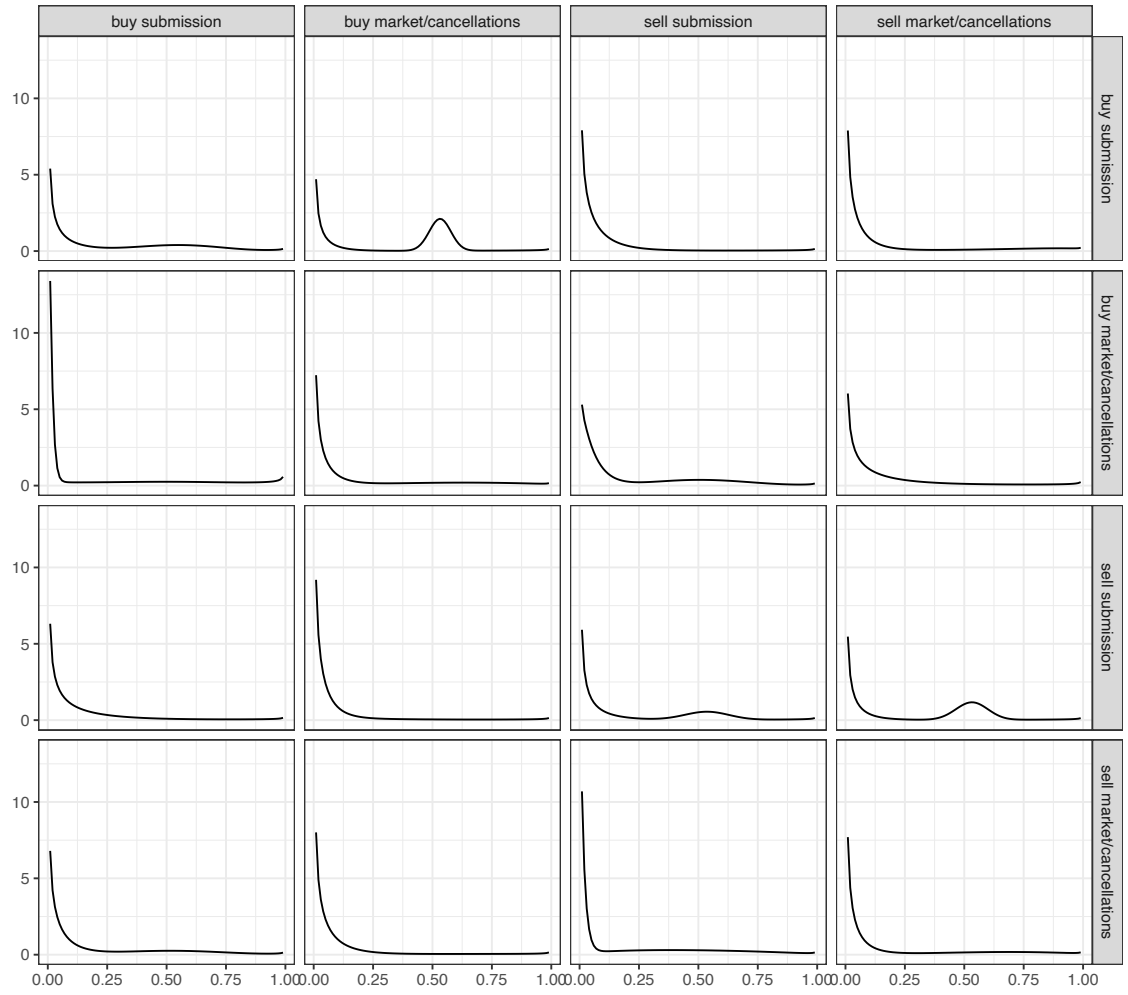

Figure 4.5: Estimated density plots from the MCMC algorithm of the cross-dimensional inter-arrival times (within one second) among the four dimensions.

Chapter 5

FUTURE DIRECTIONS

We begin this chapter with a discussion of the limitations and potential improvement directions of the methods related to previous chapters. After that, we will briefly discuss other future potential directions for research.

In Chapter 2, we introduced BARTSIMP, a flexible spatial prediction framework. BARTSIMP works best when both the spatial and covariate components contribute strongly to the modeling outcome. If the covariate signals are very weak, the prediction performance for BARTSIMP is poor compared to spatial-only methods including the SPDE model, as shown by the simulation results in Section 2.5.2. If the spatial component is very weak, BARTSIMP performs well in terms of point prediction compared to other methods like BART, but the computation is very slow (for example, see the computation results in Section 2.6.2). Furthermore, as a result of the MCMC routine in which each of the covariate variables is sequentially updated, the computation complexity in the number of dimensions grows linearly with respect to the number of covariates. In Chapter 2, we focused on simluation and application examples with a few (no more than 10) variables, a potential extension could be focused on the performance of BARTSIMP under high-dimensional settings, with potential sparse structures in the covariates (i.e. only a few variables are significantly contributing to the outcome, for example, see [78]).

In Chapters 3 and 4, we studied several computational approaches for MHP models and proposed MHP-DDP, a modeling and estimation framework that flexibly models the shapes of excitation kernels. While several improvements on the scalability of the computation methods have been proposed in Chapter 3, several challenges remain. For example, the time complexity of all three algorithms (the stochastic gradient EM, variational inference and Langevin dynamics) are quadratic in terms of number of dimensions. At the same time, the number of excitation kernels also grow quadratically with respect to the number

of dimensions. To further improve scalability when the number of dimension is large, it might be of interest to introduce sparsity structure in the number of kernels (for example, the low-rank MHP proposed by [7]).

To warrant the adequate use of MHPDDP, it is important to carry out analysis procedures including exploratory data analysis and model diagnosis. For example, an important motivation for using the model could be that there are clear non-unimodal structures and across-group difference in the empirical distributions of interarrival times across dimensions (for example, see Section 3.5) An alternative technique is to generate the quantile-quantile plots for the posterior distribution of inter-event times, according to the time-rescaling theorem [33].

The rest of this chapter of the dissertation provides a brief overview of the potential future works, beyond the scope of the previous chapters. Moving beyond statistical learning tasks including prediction and probability density estimation, which have been covered in previous chapters of this dissertation, the author would like to extend the scope of research towards applications of Bayesian nonparametric mixture models in clustering problems. Mixture models have been widely used for modeling heterogeneous population and have found applications in statistical problems including density estimation, classification and clustering. Due to their flexibility in approximating generic functions with little additional assumptions, Bayesian mixture models have been widely applied to population density estimation problems in ecology [80], epidemiology [107] and finance [76]. Well-established theoretical properties have also been studied on the asymptotic optimal estimation procedures for Bayesian nonparametric mixture models [57, 58]. Moreover, in a mixture model, each observation can be interpreted as a member of a latent group in the population. As a result, latent class labels can be assigned to the observations, and the model provides a probabilistic framework for supervised classification problems [66, 52, 148]. It has been applied to areas in political science [147] and geoscience [153]. When the true labels of the observations are unknown, mixture models can also be applied to clustering problems, with applications in. However, clustering problems are less well-defined for mixture models, compared to density estimation and classification problems. Specifically, there are a few common fundamental questions in mixture model clustering that remain open and require

careful consideration: (1) Is it possible to correctly estimate the number of clusters, when the true data come from a mixture of subpopulations? (2) What happens when the model component is misspecified? (3) Under the misspecified regime, is it possible to improve the estimation and inference through posterior processing algorithms?

Many methods have been proposed to choose an adequate number of mixture components for finite mixture models, including maximum likelihood estimation [88], the likelihood ratio test method [69, 97] and penalized likelihood estimation methods including the Bayesian information criteria (BIC) [83]. For these methods, the estimation of the number of mixture components is treated as a model selection problem. Once the number of mixtures is fixed, a separate estimation procedure can be used to conduct estimation and inference on the selected model. Alternatively, infinite mixture models combine these two steps together, allowing the potential mixtures on the model to be infinite, therefore letting the data flexibly 'decide' how many clusters are to be observed. From a Bayesian perspective, a Bayesian nonparametric (BNP) mixture model places a prior on the space of the mixing measures that corresponds to an infinite mixture model. Well known examples of such BNP infinite mixture models include the famous Dirichlet process (DP) [4, 96, 48] mixture model. However, many such models do not consistently estimate the number of mixture components. [109] and [110] have shown that the DP mixture and the Pitman-Yor process mixtures lead to inconsistent estimates of the true number of mixture components, when the data are generated from a mixture of finite components. An intuition for this is that BNP models like DP and Pitman-Yor mixtures place priors on mixture models with infinite components. As a result, the number of mixture components is inherently 'misspecified' under these models. Also, we note that consistency in population density generally does not imply consistency in number of mixtures (see, for example, [88]). Thus, even the DPM guarantees asymptotic optimal estimation for population density [57, 58], it does not lead to consistent estimation for mixture components. From an empirical perspective, as the number of observations fitted to a BNP model tends to infinity, the DP prior encourages new mixture components to appear. Such property can be demonstrated by the Polya-Urn scheme for Dirichlet process prior [15]. Alternatively, [141] studied the asymptotic behavior of the posterior distributions in overfitted finite mixture models. Under certain regular-

ity conditions, [141] showed that the redundant components are 'emptied out' with their corresponding posterior mixture weights converging to zero. From a practical standpoint, it is possible to 'correct' the number of mixtures from posterior samples of DPM through post-processing algorithms [64], under certain conditions. However, we again note that this does not imply consistent estimation of mixture components, as the redundant component being 'emptied out' is a result of model regularization rather than selection (i.e. we are still fitting an overfitted model). By contrast, the mixture of finite mixtures model (MFM, see [134, 111]) assumes that the population can be represented by a finite number of mixture components, and conducts model selection by explicitly placing a prior on the number of mixture components. Compared to DPM, MFM consistently estimates the number of mixture components under minimal identifiability assumptions [121].

Consistency result is contingent upon correct specification of mixture kernels. A key reference that studies the posterior behavior of the mixing measure under kernel misspecification is [64], which states that the posterior contracts to an approximation to the true mixing measures and provided contraction rates under certain conditions. [108] proposed coarsened posteriors as to provide robustness to slight model misspecifications by tempering the model likelihood. Nonetheless, it should be noted that it can be difficult to properly interpret the kernel parameters when the kernel is misspecified [111, 64]. An approach that deals with misspecification is to represent the kernel itself as mixtures. The idea was first explored by [66] under a supervised classification setting, where each class can be represented by a mixture of Gaussians. Under unsupervised settings, [5] proposed nonparametric mixutre modeling (NPMIX), a two-stage approach to cover the true components by fitting overfitted mixtures, and then group the mixtures together to form a larger mixture. [5] provided conditions including identifiability and separability where the algorithm works.

We would like to address the inconsistency issue under misspecified kernels through two approaches. The first approach we propose is a two-level Bayesian nonparametric model (MFM-DPM), where the model can be seen as a mixture of finite mixtures, where each of these mixtures is in turn a Dirichlet process mixture of Gaussian distributions. Due to its flexibility, simiilar models with two-level structure has been applied to classification problems [40, 10]. Intuitionally, when MFM-DPM is fitted to data coming from a population

with non-Gaussian subpopulation structures, the DPM components aim to accurately estimate each of the subpopulations, while MFM provides inference on the number of mixture components.

The second approach we propose is a post-processing algorithm inspired by the [5]. Especially, we would like to incorporate the frequentist 'mixtures-merging' procedure into Bayesian workflow. Furthermore, we would like to extend the usage of NPMIX to cases where the number of mixture components is unknown, which is a much more natural assumption. We hope these two approaches may provide theoretical and modeling insights under kernel misspecification.

BIBLIOGRAPHY

[1] Frédéric Abergel and Aymen Jedidi. Long-time behavior of a Hawkes process–based limit order book. *Siam Journal on Financial Mathematics*, 6(1):1026–1043, 2015.

[2] Milton Abramowitz and Irene A Stegun. *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*, volume 55. US Government printing office, 1964.

[3] Aurélien Alfonsi, Antje Fruth, and Alexander Schied. Optimal execution strategies in limit order books with general shape functions. *Quantitative Finance*, 10(2):143–157, 2010.

[4] Charles E Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, pages 1152–1174, 1974.

[5] Bryon Aragam, Chen Dan, Eric P. Xing, and Pradeep Ravikumar. Identifiability of nonparametric mixture models and Bayes optimal clustering. *The Annals of Statistics*, 48(4):2277–2302, 2020.

[6] Emmanuel Bacry, Martin Bompaire, Stéphane Gaiffas, and Jean-Francois Muzy. Sparse and low-rank multivariate Hawkes processes. *Journal of Machine Learning Research*, 21(50):1–32, 2020.

[7] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005, 2015.

[8] Emmanuel Bacry and Jean-François Muzy. First-and second-order statistics characterization of Hawkes processes and non-parametric estimation. *Ieee Transactions on Information Theory*, 62(4):2184–2202, 2016.

[9] Sudipto Banerjee. Finite population survey sampling: An unapologetic Bayesian perspective. *Sankhya a*, 86:1–30, 2024.

[10] Francesco Bartolucci. Clustering univariate observations via mixtures of unimodal normal mixtures. *Journal of Classification*, 22(2), 2005.

[11] George E Battese, Rachel M Harter, and Wayne A Fuller. An error-components model for prediction of county crop areas using survey and satellite data. *Journal of the American Statistical Association*, 83(401):28–36, 1988.

[12] Penny Bilton, Geoff Jones, Siva Ganesh, and Steve Haslett. Classification trees for poverty mapping. *Computational Statistics & Data Analysis*, 115:53–66, 2017.

[13] Christopher M. Bishop. *Pattern Recognition and Machine Learning (information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[14] Christopher M Bishop and Nasser M Nasrabadi. *Pattern Recognition and Machine Learning*. Springer, 2006.

[15] David Blackwell and James B MacQueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1(2):353–355, 1973.

[16] David M. Blei and Michael I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121 – 143, 2006.

[17] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: a review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[18] Anna Bonnet, Miguel Martinez Herrera, and Maxime Sangnier. Inference of multivariate exponential Hawkes processes with inhibition and application to neuronal activity. *Statistics and Computing*, 33(4):91, 2023.

[19] Claudio Bosco, Victor Alegana, Tomas Bird, Carla Pezzulo, Linus Bengtsson, Alessandro Sorichetta, Jessica Steele, G Hornby, C Ruktanonchai, N Ruktanonchai, et al. Exploring the high-resolution mapping of gender-disaggregated development indicators. *Journal of the Royal Society Interface*, 14(129):20160825, 2017.

[20] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[21] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[22] Pierre Brémaud and Laurent Massoulié. Stability of nonlinear Hawkes processes. *The Annals of Probability*, pages 1563–1588, 1996.

[23] Marshall Burke, Anne Driscoll, David B Lobell, and Stefano Ermon. Using satellite imagery to understand and promote sustainable development. *Science*, 371(6535):eabe8628, 2021.

[24] Roy Burstein, Nathaniel J Henry, Michael L Collison, Laurie B Marczak, Amber Sligar, Stefanie Watson, Neal Marquez, Mahdieh Abbasalizad-Farhangi, Masoumeh Abbasi, Foad Abd-Allah, et al. Mapping 123 million neonatal, infant and child deaths between 2000 and 2017. *Nature*, 574(7778):353–358, 2019.

[25] Olivier Cappe and Eric Moulines. On-line expectation maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (statistical Methodology)*, 71(3):593–613, 2009.

[26] Alec M Chan-Golston, Sudipto Banerjee, and Mark S Handcock. Bayesian inference for finite populations under spatial process settings. *Environmetrics*, 31(3):e2606, 2020.

[27] Jianfei Chen, Jun Zhu, Yee Whye Teh, and Tong Zhang. Stochastic expectation maximization with variance reduction. *Advances in Neural Information Processing Systems*, 31, 2018.

[28] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, pages 1683–1691. PMLR, 2014.

[29] Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998.

[30] Hugh A Chipman, Edward I George, and Robert E McCulloch. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

[31] Rama Cont. Statistical modeling of high-frequency financial data. *Ieee Signal Processing Magazine*, 28(5):16–25, 2011.

[32] Rama Cont and Marvin S Müller. A stochastic partial differential equation model for limit order book dynamics. *Siam Journal on Financial Mathematics*, 12(2):744–787, 2021.

[33] Daryl J Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes. Volume Ii: General Theory and Structure*. Springer, 2008.

[34] Abhirup Datta, Sudipto Banerjee, Andrew O Finley, and Alan E Gelfand. On nearest-neighbor gaussian process models for massive spatial data. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(5):162–171, 2016.

[35] Molly Margaret Davies and Mark J Van Der Laan. Optimal spatial prediction using ensemble machine learning. *The International Journal of Biostatistics*, 12(1):179–201, 2016.

[36] Ranadeep Daw and Christopher K Wikle. REDS: Random ensemble deep spatial prediction. *Environmetrics*, 34(1):e2780, 2023.

[37] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (methodological)*, 39(1):1–22, 1977.

[38] Isabella Deutsch and Gordon J Ross. Bayesian estimation of multivariate Hawkes processes with inhibition and sparsity. *Arxiv Preprint Arxiv:2201.05009*, 2022.

[39] Ruben Dezeure, Peter Bühlmann, Lukas Meier, and Nicolai Meinshausen. High-dimensional inference: confidence intervals, p-values and R-software hdi. *Statistical Science*, 30(4):533–558, 2015.

[40] Marco Di Zio, Ugo Guarnera, and Roberto Rocci. A mixture of mixture models for a classification problem: the unity measure error. *Computational Statistics & Data Analysis*, 51(5):2573–2585, 2007.

[41] Peter J Diggle and Emanuele Giorgi. *Model-Based Geostatistics for Global Public Health: Methods and Applications*. CRC Press, 2019.

[42] Sophie Donnet, Vincent Rivoirard, and Judith Rousseau. Nonparametric Bayesian estimation for multivariate Hawkes processes. *The Annals of Statistics*, 48(5):2698 – 2727, 2020.

[43] Ian L Dryden and Kanti V Mardia. *Statistical Shape Analysis: With Applications in R*, volume 995. John Wiley & Sons, 2016.

[44] Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21th Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pages 219–228, 2015.

[45] Dirk Eddelbuettel. *Seamless R and C++ Integration with Rcpp*. Springer, 2013.

[46] Dirk Eddelbuettel and James Joseph Balamuta. Extending R with C++: a brief introduction to Rcpp. *The American Statistician*, 72(1):28–36, 2018.

[47] Dirk Eddelbuettel and Romain François. Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software*, 40:1–18, 2011.

[48] Michael D Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[49] Robert E Fay and Roger A Herriot. Estimates of income for small places: an application of James-Stein procedures to census data. *Journal of the American Statistical Association*, 74(366a):269–277, 1979.

[50] Andrew O Finley, Hans-Erik Andersen, Chad Babcock, Bruce D Cook, Douglas C Morton, and Sudipto Banerjee. Models to support forest inventory and small area estimation using sparsely sampled LiDAR: A case study involving G-LiHT LiDAR in Tanana, Alaska. *Journal of Agricultural, Biological and Environmental Statistics*, 29:1–28, 2024.

[51] Nick Foti, Jason Xu, Dillon Laird, and Emily Fox. Stochastic variational inference for hidden Markov models. *Advances in Neural Information Processing Systems*, 27, 2014.

[52] Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

[53] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

[54] Geir-Arne Fuglstad, Daniel Simpson, Finn Lindgren, and HÃvard Rue. Constructing priors that penalize the complexity of Gaussian random fields. *Journal of the American Statistical Association*, 114(525):445–452, 2019.

[55] Reinhard Furrer, Marc G Genton, and Douglas Nychka. Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3):502–523, 2006.

[56] Stefanos Georganos, Tais Grippa, Assane Niang Gadiaga, Catherine Linard, Moritz Lennert, Sabine Vanhuysse, Nicholus Mboga, Eléonore Wolff, and Stamatis Kalogirou. Geographical random forests: a spatial extension of the random forest algorithm to address spatial heterogeneity in remote sensing and population modelling. *Geocarto International*, 36(2):121–136, 2021.

[57] S. Ghosal, J. K. Ghosh, and R. V. Ramamoorthi. Posterior consistency of Dirichlet mixtures in density estimation. *The Annals of Statistics*, 27(1):143 – 158, 1999.

[58] Subhashis Ghosal and Aad van der Vaart. Posterior convergence rates of Dirichlet mixtures at smooth densities. *The Annals of Statistics*, 35(2):697 – 723, 2007.

[59] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[60] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.

[61] Virgilio Gómez-Rubio and Håvard Rue. Markov chain Monte Carlo with the integrated nested Laplace approximation. *Statistics and Computing*, 28(5):1033–1051, 2018.

[62] Prem K Gopalan, Sean Gerrish, Michael Freedman, David Blei, and David Mimno. Scalable inference of overlapping communities. *Advances in Neural Information Processing Systems*, 25, 2012.

[63] Martin D Gould, Mason A Porter, Stacy Williams, Mark McDonald, Daniel J Fenn, and Sam D Howison. Limit order books. *Quantitative Finance*, 13(11):1709–1742, 2013.

[64] Aritra Guha, Nhat Ho, and XuanLong Nguyen. On posterior contraction of parameters and interpretability in Bayesian mixture modeling. *Bernoulli*, 27(4):2159–2188, 2021.

[65] Niels Richard Hansen, Patricia Reynaud-Bouret, and Vincent Rivoirard. Lasso and probabilistic inequalities for multivariate point processes. *Bernoulli*, 21(1):83 – 143, 2015.

[66] Trevor Hastie and Robert Tibshirani. Discriminant analysis by Gaussian mixtures.

Journal of the Royal Statistical Society Series B: Statistical Methodology, 58(1):155–176, 1996.

[67] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[68] Alan G Hawkes and David Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, 11(3):493–503, 1974.

[69] Jogi Henna. On estimating of the number of constituents of a finite mixture of continuous distributions. *Annals of the Institute of Statistical Mathematics*, 37:235–240, 1985.

[70] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[71] Andrew J Holbrook, Xiang Ji, and Marc A Suchard. From viral evolution to spatial contagion: a biologically modulated Hawkes model. *Bioinformatics*, 38(7):1846–1856, 2022.

[72] Andrew J Holbrook, Charles E Loeffler, Seth R Flaxman, and Marc A Suchard. Scalable Bayesian inference for self-excitatory stochastic processes applied to big american gunfire data. *Statistics and Computing*, 31(1):1–15, 2021.

[73] Ruihong Huang and Tomas Polak. Lobster: Limit order book reconstruction system. *Available at Ssrn 1977207*, 2011.

[74] Aliaksandr Hubin and Geir Storvik. Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA). *Arxiv Preprint Arxiv:1611.01450*, 2016.

[75] Hemant Ishwaran and Lancelot F James. Approximate Dirichlet process computing in finite normal mixtures: smoothing and prior information. *Journal of Computational and Graphical Statistics*, 11(3):508–532, 2002.

[76] A. Jasra, C. C. Holmes, and D. A. Stephens. Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling. *Statistical Science*, 20(1):50 – 67, 2005.

[77] Neal Jean, Marshall Burke, Michael Xie, W Matthew Alampay Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

[78] Seonghyun Jeong and Veronika Rockova. The art of bart: Minimax optimality over nonhomogeneous smoothness in high dimension. *Journal of Machine Learning Research*, 24(337):1–65, 2023.

[79] Alex Ziyu Jiang and Abel Rodriguez. Improvements on scalable stochastic Bayesian inference methods for multivariate Hawkes process. *Statistics and Computing*, 34(2):85, 2024.

[80] Liana N Joseph, Ché Elkin, Tara G Martin, and Hugh P Possingham. Modeling abundance using n-mixture models: the importance of considering ecological mechanisms. *Ecological Applications*, 19(3):631–642, 2009.

[81] Gashu Workneh Kassie and Demeke Lakew Workie. Exploring the association of anthropometric indicators for under-five children in Ethiopia. *Bmc Public Health*, 19(1):1–6, 2019.

[82] Cari G Kaufman, Mark J Schervish, and Douglas W Nychka. Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association*, 103(484):1545–1555, 2008.

[83] Christine Keribin. Consistent estimation of the order of mixture models. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 49–66, 2000.

[84] Mohammad Khan and Wu Lin. Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models. In *Artificial Intelligence and Statistics*, pages 878–887. PMLR, 2017.

[85] John FC Kingman. Random discrete distributions. *Journal of the Royal Statistical Society: Series B (methodological)*, 37(1):1–15, 1975.

[86] Leonhard Knorr-Held and Håvard Rue. On block updating in Markov random field models for disease mapping. *Scandinavian Journal of Statistics*, 29(4):597–614, 2002.

[87] Rico Krueger, Prateek Bansal, and Prasad Buddhavarapu. A new spatial count data model with Bayesian additive regression trees for accident hot spot identification. *Accident Analysis and Prevention*, 144:105623, 2020.

[88] Brian G Leroux. Consistent estimation of a mixing distribution. *The Annals of Statistics*, pages 1350–1360, 1992.

[89] James P. LeSage and R. Kelley Pace. A matrix exponential spatial specification. *Journal of Econometrics*, 140(1):190–214, 2007. Analysis of spatially dependent data.

[90] Erik Lewis and George Mohler. A nonparametric EM algorithm for multiscale Hawkes processes. *Journal of Nonparametric Statistics*, 01 2011.

[91] Scott W Linderman and Ryan P Adams. Scalable Bayesian inference for excitatory point process networks. *Arxiv Preprint Arxiv:1507.03228*, 2015.

[92] Scott W Linderman, Yixin Wang, and David M Blei. Bayesian inference for latent Hawkes processes. *Advances in Neural Information Processing Systems*, 2017.

[93] Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B*, 73(4):423–498, 2011.

[94] Johan Lindström, Adam A Szpiro, Paul D Sampson, Assaf P Oron, Mark Richards, Tim V Larson, and Lianne Sheppard. A flexible spatio-temporal model for air pollution with spatial and spatio-temporal covariates. *Environmental and Ecological Statistics*, 21:411–433, 2014.

[95] Thomas Josef Liniger. *Multivariate Hawkes processes*. PhD thesis, ETH Zurich, 2009.

[96] Albert Y Lo. On a class of Bayesian nonparametric estimates: I. density estimates. *The Annals of Statistics*, pages 351–357, 1984.

[97] Yungtai Lo, Nancy R Mendell, and Donald B Rubin. Testing the number of components in a normal mixture. *Biometrika*, 88(3):767–778, 2001.

[98] A. Logothetis and V. Krishnamurthy. Expectation maximization algorithms for MAP estimation of jump Markov linear systems. *Ieee Transactions on Signal Processing*, 47(8):2139–2156, 1999.

[99] Zhanyu Ma and Arne Leijon. Bayesian estimation of Beta mixture models with variational inference. *Ieee Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2160–2173, 2011.

[100] Peter M Macharia, Emanuele Giorgi, Pamela N Thuranira, Noel K Joseph, Benn Sartorius, Robert W Snow, and Emelda A Okiro. Sub national variation and inequalities in under-five mortality in Kenya since 1965. *Bmc Public Health*, 19(1):1–12, 2019.

[101] Noa Malem-Shinitski, César Ojeda, and Manfred Opper. Variational Bayesian inference for nonlinear Hawkes process with Gaussian process self-effects. *Entropy*, 24(3):356, 2022.

[102] Dean Markwick. *Bayesian Nonparametric Hawkes Processes with Applications*. PhD thesis, UCL (University College London), 2020.

[103] David Marsan and Olivier Lengline. Extending earthquakes' reach through cascading. *Science*, 319(5866):1076–1079, 2008.

[104] Thiago G Martins, Daniel Simpson, Finn Lindgren, and Håvard Rue. Bayesian computing with INLA: new features. *Computational Statistics and Data Analysis*, 67:68–83, 2013.

[105] Hongyuan Mei and Jason M Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. In I. Guyon, U. Von Luxburg, S. Bengio,

H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[106] Zuguo Mei and Laurence M Grummer-Strawn. Standard deviation of anthropometric Z-scores as a data quality assessment tool using the 2006 WHO growth standards: a cross country analysis. *Bulletin of the World Health Organization*, 85:441–448, 2007.

[107] AF Militino, MD Ugarte, and CB Dean. The use of mixture models for identifying high risks in disease mapping. *Statistics in Medicine*, 20(13):2035–2049, 2001.

[108] Jeffrey W Miller and David B Dunson. Robust Bayesian inference via coarsening. *Journal of the American Statistical Association*, 2019.

[109] Jeffrey W Miller and Matthew T Harrison. A simple example of Dirichlet process mixture inconsistency for the number of components. *Advances in Neural Information Processing Systems*, 26, 2013.

[110] Jeffrey W Miller and Matthew T Harrison. Inconsistency of Pitman-Yor process mixtures for the number of components. *The Journal of Machine Learning Research*, 15(1):3333–3370, 2014.

[111] Jeffrey W Miller and Matthew T Harrison. Mixture models with a prior on the number of components. *Journal of the American Statistical Association*, 113(521):340–356, 2018.

[112] Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, pages 1023–1032, 1988.

[113] George Mohler. Modeling and estimation of multi-source clustering in crime and security data. *The Annals of Applied Statistics*, pages 1525–1539, 2013.

[114] Peter Müller and Riten Mitra. Bayesian nonparametric inference–why and how. *Bayesian Analysis (online)*, 8(2):10–1214, 2013.

[115] Peter Müller, Fernando Quintana, and Gary Rosner. A method for combining inference across related nonparametric Bayesian models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 66(3):735–749, 2004.

[116] Peter Müller, Ya-Chen Tina Shih, and Song Zhang. A spatially-adjusted Bayesian additive regression tree model to merge two datasets. *Bayesian Analysis*, 2(3):611–633, 2007.

[117] Ioane Muni Toke. The order book as a queueing system: Average depth and influence of the size of limit orders. *Quantitative Finance*, 15(5):795–808, 2015.

[118] Radford Neal. MCMC Using Hamiltonian Dynamics. In *Handbook of Markov Chain Monte Carlo*, pages 113–162. CRC Press, 2011.

[119] Christopher Nemeth and Paul Fearnhead. Stochastic gradient Markov chain Monte Carlo. *Journal of the American Statistical Association*, 116(533):433–450, 2021.

[120] Maximilian Nickel and Matthew Le. Learning multivariate Hawkes processes at scale. *Arxiv Preprint Arxiv:2002.12501*, 2020.

[121] Agostino Nobile. *Bayesian analysis of finite mixture distributions*. Carnegie Mellon University, 1994.

[122] Douglas Nychka, Soutir Bandyopadhyay, Dorit Hammerling, Finn Lindgren, and Stephan Sain. A multiresolution gaussian process model for the analysis of large spatial datasets. *Journal of Computational and Graphical Statistics*, 24(2):579–599, 2015.

[123] Yosihiko Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association*, 83(401):9–27, 1988.

[124] Aaron Osgood-Zimmerman, Anoushka I Millear, Rebecca W Stubbs, Chloe Shields, Brandon V Pickering, Lucas Earl, Nicholas Graetz, Damaris K Kinyoki, Sarah E Ray,

Samir Bhatt, et al. Mapping child growth failure in Africa between 2000 and 2015. *Nature*, 555(7694):41–47, 2018.

[125] Aaron Osgood-Zimmerman and Jon Wakefield. A statistical review of template model builder: a flexible tool for spatial modelling. *International Statistical Review*, 91:318–342, 2023.

[126] Tohru Ozaki. Maximum likelihood estimation of Hawkes' self-exciting point processes. *Annals of the Institute of Statistical Mathematics*, 31:145–155, 1979.

[127] Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using Pólya–Gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013.

[128] Fernando A Quintana, Peter Müller, Alejandro Jara, and Steven N MacEachern. The dependent Dirichlet process and related models. *Statistical Science*, 37(1):24–41, 2022.

[129] Marcello Rambaldi, Emmanuel Bacry, and Fabrizio Lillo. The role of volume in order book dynamics: a multivariate Hawkes process analysis. *Quantitative Finance*, 17(7):999–1020, 2017.

[130] John NK Rao and Isabel Molina. *Small Area Estimation*. John Wiley & Sons, 2015.

[131] Jakob Gulddahl Rasmussen. Bayesian inference for Hawkes processes. *Methodology and Computing in Applied Probability*, 15(3):623–642, 2013.

[132] Zhoupeng Ren, Jun Zhu, Yanfang Gao, Qian Yin, Maogui Hu, Li Dai, Changfei Deng, Lin Yi, Kui Deng, Yanping Wang, Xiaohong Li, and Jinfeng Wang. Maternal exposure to ambient PM10 during pregnancy increases the risk of congenital heart defects: Evidence from machine learning models. *Science of the Total Environment*, 630:1–10, 2018.

[133] Patricia Reynaud-Bouret and Sophie Schbath. Adaptive estimation for Hawkes processes; application to genome analysis. *The Annals of Statistics*, 38(5):2781 – 2822, 2010.

[134] Sylvia Richardson and Peter J Green. On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 59(4):731–792, 1997.

[135] Marian-Andrei Rizoiu, Young Lee, Swapnil Mishra, and Lexing Xie. Hawkes processes for events in social media. In Shih-Fu Chang, editor, *Frontiers of Multimedia Research*, pages 191–218. Morgan & Claypool Publishers, 2017.

[136] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951.

[137] Abel Rodriguez and David B Dunson. Nonparametric Bayesian models through probit stick-breaking processes. *Bayesian Analysis (Online)*, 6(1), 2011.

[138] Abel Rodriguez, David B Dunson, and Alan E Gelfand. The nested Dirichlet process. *Journal of the American Statistical Association*, 103(483):1131–1154, 2008.

[139] Abel Rodríguez and Peter Müller. *Nonparametric Bayesian inference*. NSF-CBMS Regional Conference Series in Probability and Statistics. Institute of Mathematical Statistics, 2013.

[140] Abel Rodríguez, Ziwei Wang, and Athanasios Kottas. Assessing systematic risk in the S&P500 index between 2000 and 2011: A Bayesian nonparametric approach. *The Annals of Applied Statistics*, pages 527–552, 2017.

[141] Judith Rousseau and Kerrie Mengersen. Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(5):689–710, 2011.

[142] Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B*, 71(2):319–392, 2009.

[143] Frederic Paik Schoenberg. Facilitated estimation of etas. *Bulletin of the Seismological Society of America*, 103(1):601–605, 2013.

[144] Matthias Seeger. Gaussian processes for machine learning. *International Journal of Neural Systems*, 14(02):69–106, 2004.

[145] Didier Sornette and S Utkin. Limits of declustering methods for disentangling exogenous from endogenous events in time series with foreshocks, main shocks, and aftershocks. *Physical Review E—statistical, Nonlinear, and Soft Matter Physics*, 79(6):061110, 2009.

[146] Charles Spanbauer and Rodney Sparapani. Nonparametric machine learning for precision medicine with longitudinal clinical trials and Bayesian additive regression trees with mixed models. *Statistics in Medicine*, 40(11):2665–2691, 2021.

[147] Arthur Spirling and Kevin Quinn. Identifying intraparty voting blocs in the uk house of commons. *Journal of the American Statistical Association*, 105(490):447–457, 2010.

[148] Santosh Srivastava, Maya R Gupta, and Béla A Frigyik. Bayesian quadratic discriminant analysis. *Journal of Machine Learning Research*, 8(6), 2007.

[149] Deborah Sulem, Vincent Rivoirard, and Judith Rousseau. Bayesian estimation of nonlinear Hawkes process. *Arxiv Preprint Arxiv:2103.17164*, 2021.

[150] Deborah Sulem, Vincent Rivoirard, and Judith Rousseau. Scalable variational Bayes methods for Hawkes processes. *Arxiv Preprint Arxiv:2212.00293*, 2022.

[151] Yaoyuan Vincent Tan and Jason Roy. Bayesian additive regression trees and the general BART model. *Statistics in Medicine*, 38(25):5048–5069, 2019.

[152] Yee Teh, Michael Jordan, Matthew Beal, and David Blei. Sharing clusters among related groups: Hierarchical Dirichlet processes. *Advances in Neural Information Processing Systems*, 17, 2004.

[153] Dieu Tien Bui and Nhat-Duc Hoang. A Bayesian framework based on a Gaussian mixture model and radial-basis-function fisher discriminant analysis (baygmmkda v1. 1) for spatial prediction of floods. *Geoscientific Model Development*, 10(9):3391–3409, 2017.

[154] Warren S Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[155] C Edson Utazi, Julia Thorley, Victor A Alegana, Matthew J Ferrari, Saki Takahashi, C Jessica E Metcalf, Justin Lessler, and Andrew J Tatem. High resolution age-structured mapping of childhood vaccination coverage in low and middle income countries. *Vaccine*, 36(12):1583–1591, 2018.

[156] Vestine Uwiringiyimana, Frank Osei, Sherif Amer, and Antonie Veldkamp. Bayesian geostatistical modelling of stunting in Rwanda: risk factors and spatially explicit residual stunting burden. *Bmc Public Health*, 22(1):1–14, 2022.

[157] Alejandro Veen and Frederic P Schoenberg. Estimation of space-time branching process models in seismology using an EM-type algorithm. *Journal of the American Statistical Association*, 103(482):614–624, 2008.

[158] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[159] Jon Wakefield, Geir-Arne Fuglstad, Andrea Riebler, Jessica Godwin, Katie Wilson, and Samuel J Clark. Estimating under-five mortality in space and time in a developing world context. *Statistical Methods in Medical Research*, 28(9):2614–2634, 2019.

[160] Stephen Walker, Paul Damien, and Peter Lenk. On priors with a Kullback–Leibler property. *Journal of the American Statistical Association*, 99(466):404–408, 2004.

[161] Chong Wang and David M Blei. Variational inference in nonconjugate models. *Journal of Machine Learning Research*, 2013.

[162] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (icml-11)*, pages 681–688, 2011.

[163] Yuefeng Wu and Subhashis Ghosal. Kullback Leibler property of kernel mixture priors in Bayesian density estimation. *Electronic Journal of Statistics*, 2(none):298 – 331, 2008.

[164] Hongteng Xu and Hongyuan Zha. A Dirichlet mixture model of Hawkes processes for event sequence clustering. *Advances in Neural Information Processing Systems*, 30, 2017.

[165] Mojtaba Zeraatpisheh, Younes Garosi, Hamid Reza Owliaie, Shamsollah Ayoubi, Ruhollah Taghizadeh-Mehrjardi, Thomas Scholten, and Ming Xu. Improving the spatial prediction of soil organic carbon using environmental covariates selection: a comparison of a group of environmental covariates. *Catena*, 208:105723, 2022.

[166] Rui Zhang, Christian Walder, and Marian-Andrei Rizoiu. Variational inference for sparse Gaussian process modulated Hawkes process. In *Proceedings of the Aaai Conference on Artificial Intelligence*, volume 34, pages 6803–6810, 2020.

[167] Rui Zhang, Christian Walder, Marian-Andrei Rizoiu, and Lexing Xie. Efficient non-parametric Bayesian Hawkes processes. *Arxiv Preprint Arxiv:1810.03730*, 2018.

[168] Feng Zhou, Quyu Kong, Yixuan Zhang, Cheng Feng, and Jun Zhu. Nonlinear Hawkes processes in time-varying system. *Arxiv Preprint Arxiv:2106.04844*, 2021.

[169] Feng Zhou, Zhidong Li, Xuhui Fan, Yang Wang, Arcot Sowmya, and Fang Chen. Efficient inference for nonparametric Hawkes processes using auxiliary latent variables. *Journal of Machine Learning Research*, 2020.

[170] Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional Hawkes processes. In *International Conference on Machine Learning*, pages 1301–1309. PMLR, 2013.