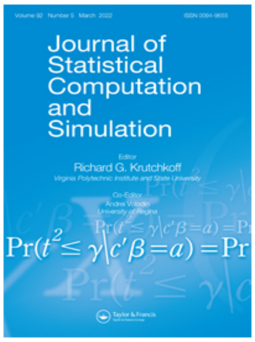
Volume 92 Number 5 March 2022
ISSN 0094-9655
Journal of Statistical Computation and Simulation
Editor
Richard G. Krutchkoff
Virginia Polytechnic Institute and State University
Co-Editor
Andrei Volodin
University of Regina
$\Pr(t^2 \leq \gamma | c'\beta = a) = \Pr$
Taylor & Francis

Taylor & Francis
Taylor & Francis Group

Journal of Statistical Computation and Simulation

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/gscs20

Online Bayesian learning for mixtures of spatial spline regressions with mixed effects

Shufei Ge, Shijia Wang, Farouk S. Nathoo & Liangliang Wang

To cite this article: Shufei Ge, Shijia Wang, Farouk S. Nathoo & Liangliang Wang (2021): Online Bayesian learning for mixtures of spatial spline regressions with mixed effects, Journal of Statistical Computation and Simulation, DOI: 10.1080/00949655.2021.2002329

To link to this article: https://doi.org/10.1080/00949655.2021.2002329

Published online: 24 Nov 2021.

JOURNAL OF STATISTICAL COMPUTATION AND SIMULATION
https://doi.org/10.1080/00949655.2021.2002329

Online Bayesian learning for mixtures of spatial spline regressions with mixed effects

Shufei Ge [a], Shijia Wang [b], Farouk S. Nathoo [c] and Liangliang Wang [d]

[a]Institute of Mathematical Sciences, ShanghaiTech University, Shanghai, People's Republic of China; [b]School of Statistics and Data Science, LPMC& KLMDASR, Nankai University, Tianjin, People's Republic of China; [c]Department of Mathematics and Statistics, University of Victoria, Victoria, Canada; [d]Department of Statistics and Actuarial Science, Simon Fraser University, Burnaby, Canada

ABSTRACT

Classification and clustering methods based on univariate functions have been well developed. Recent work has extended the techniques to the domain of bivariate functions by incorporating the techniques based on mixtures of spatial spline regression with mixed-effects models. An Expectation Maximization (EM) algorithm is implemented to facilitate model inference. In this paper, we further extend the mixtures of spatial spline regression with mixed-effects model under the Bayesian framework to accommodate streaming image data. First, we derive a Markov chain Monte Carlo (MCMC) algorithm as an alternative approach to the EM algorithm to make inference on the model. However, MCMC is not scalable to streaming image data since it requires all observed information to update the posterior distribution of the parameters. To tackle this issue, we propose a sequential Monte Carlo (SMC) algorithm to analyse online fashion image data. The existence of model sufficient statistics improves the efficiency of the proposed online SMC algorithm. Instead of saving all batch data for inference, we only require storage of the model sufficient statistics and every data point is only used once, which is well suited for large-scale stream type data. In addition, the proposed algorithm provides an unbiased estimator of the marginal likelihood as a by-product of the approach, which can be used for model selection. Numerical experiments are used to demonstrate the effectiveness of our method. Our implementation is available at https://github.com/ShufeiGe/Online-Bayesian-learning-for-MMSRm.

ARTICLE HISTORY
Received 1 February 2021
Accepted 31 October 2021

KEYWORDS
Online learning; sequential Monte Carlo; functional data analysis

1. Introduction

Classification and clustering of random objects has received substantial attention in functional data analysis. James and Hastie [1] proposed the functional linear discriminant analysis, which can perform the classification of curves, by extending the classical method of linear discriminant analysis [2] to functional data. Motivated by radar-range problems, Hall et al. [3] developed the classification of Gaussian process regression to do real-time discrimination in the context of signal analysis based on principal coordinates analysis.

CONTACT Liangliang Wang lwa68@sfu.ca; Shijia Wang shijia_wang@nankai.edu.cn

James and Sugar [4] generalized the clustering of functional data based on mixtures of linear mixed models, and their method is particularly well suited for sparsely sampled data. Müller [5] extended generalized functional linear models to model sparse and irregular predictors for classification. Chamroukhi et al. [6] proposed the clustering of curves by combing the piecewise polynomial regression with a discrete hidden regression model to accommodate abrupt or smooth changes in curves. Cheng et al. [7] proposed an efficient parameter estimation method in a generalized partially linear additive models for both longitudinal and clustered data. The nonparametric part of the model is approximated by splines. Huang et al. [8] developed a general framework to jointly model and cluster functional trajectories.

Although not as abundant as univariate functions, bivariate functional data analysis has also been well developed as a statistical tool in surface smoothing and estimation. Matheron [9] introduced a geostatistical procedure called 'kriging', which predicts the value of a spatial process at any arbitrary location via a weighted average of available samples. Duchon [10] proposed the use of thin-plate splines as an interpolation method for surfaces and a penalty term was introduced to control the smoothness of the fitted surface. Malfait and Ramsay [11] proposed a spatial spline regression (SSR) model to deal with functional time series. Wood et al. [12] showed that conventional smoothing methods may not work well for complex domains, due to poor performance at the boundary of these domains. These authors discuss a technique known as soap film smoothing which can be applied to smoothing over difficult subregions of $\mathbb{R}^2$. Xun and Cao [13] investigated a historical functional linear model with an unknown historical forward time lag. The triangular basis functions are used to model the coefficient function. The aforementioned work has a main focus on surface smoothing and estimation for a single population rather than clustering of surfaces from several populations.

Nguyen et al. [14] proposed a new application of bivariate functions to do clustering and classification for surfaces by extending the clustering techniques of [4] to mixtures of spatial spline regression with mixed-effects model. We will refer to mixture of spatial spline regression with mixed-effects as '$\mathrm{MSSR_m}$' in what follows. Nguyen et al. [14] fitted the $\mathrm{MSSR_m}$ model using the Expectation-Maximization (EM) algorithm [15]. In the E-step the expectation of the log-likelihood function is computed over the latent variables conditioned on the observations and the current parameter estimates. In the M-step the expected log-likelihood obtained in the E-step is maximized over the model parameters. Srivastava et al. [16] developed an asynchronous distributed expectation maximization (DEM) algorithm for large-scale data sets with the divide-and-conquer technique. In DEM, the E-step is run in parallel on multiple worker processes, and the managers perform the M-step with a fraction of the results from the local expectation step.

Markov chain Monte Carlo (MCMC) [17,18] is a standard approach to make inference for latent variable model in the Bayesian framework. Besides point estimates, MCMC provides uncertainty quantification for the parameters of interest. In addition, the inclusion of prior knowledge arises naturally in the Bayesian framework. Chamroukhi [19], Chamroukhi and Nguyen [20] consider Bayesian approaches to the problem of static estimation for spatial spline regression. However, there are several constraints for an MCMC algorithm to be implemented for inference in an online problem. First of all, the storage of such big data may cause memory issues in practice. Second, the MCMC has to be rerun to update parameters when a new observation arrives, which is inefficient for dynamic problems. Finally, the marginal likelihood is expensive and challenging to compute in an MCMC algorithm.

Sequential Monte Carlo (SMC) [21–23], aka Particle Filtering, provides a sequential approximation of the posterior distribution using sampled particles (samples), and is often used to learn the distribution of latent variables. SMC has been demonstrated as a gold standard for dynamic problems, which leads to interest in implementing this method to many batch problems as an alternative to traditional Bayesian computation methods (e.g. MCMC). The sequential scheme in SMC makes online inference in latent variable models possible. Some sophisticated work has been done on mixture models using SMC [24–26]. When a new observation arrives, within this framework we only need to update the uncertainty immediately using this new observation and the low dimensional sufficient statistics we have stored, which means every observation is only used once. This is especially suitable for large-scale stream type data. In addition, another advantage of our proposed SMC algorithm over MCMC is that the marginal likelihood, which can be used for model selection, can be conveniently calculated. The posterior cluster allocation is also obtained from the algorithm.

In this paper, we build a Bayesian hierarchical model based on the $\mathrm{MSSR_m}$ as described in Section 3.1, in which we take the cluster labels as latent variables. The latent variables of the model are discrete and evolve with time. We apply the SMC scheme to learn the distribution of these latent variables and static parameters of the model. As described in Algorithm 2, we take the Resample-Propagate strategy to reduce the approximation errors since both the resample and propagate step will be informed by the new observation [24,27,28]. In addition, we include MCMC moves within the SMC algorithm to mutate particles so as to prevent the progressive degeneration [29,30]. Moreover, with the adoption of model sufficient statistics, instead of requiring storage of an entire data set, only sufficient statistics are required to summarize the data in the MCMC move, and this scheme makes the SMC algorithm computationally more efficient.

In this paper, we have three main contributions. First, we derive the full MCMC (Gibbs) sampling scheme for the $\mathrm{MSSR_m}$ model. Second, we propose an online SMC algorithm for image clustering based on the $\mathrm{MSSR_m}$ model, which allows us to conduct model inference for sequential images efficiently. Our proposed method is related to the work in [24]. Their method is applied to relatively simple mixture models. In contrast, our spatial spline mixture has a more complicated model structure, high dimensional model parameters and a large number of random effects. Third, we numerically assess the performance of our algorithm with different number of latent states (clusters) and observations.

The remainder of this article is organized as follows. We introduce the mixture of spatial spline regression model in Section 2. In Section 3, we derive the Gibbs sampling algorithm for the mixture of spatial spline regression and propose an online inference scheme for this model. In Section 4, we introduce the model selection criteria for the proposed online SMC algorithm. In Section 5, we use a numerical study to assess and compare the performance of EM, DEM, MCMC and the proposed online SMC algorithm. In Section 6, we apply the methods to hand writing recognition data and brain image magnetoencephalography (MEG) data. We list all notations in Appendix 1.

2. Mixture of spatial spline regression model

The spatial spline regression (SSR) dates back to [11,31]. Malfait and Ramsay [11] first applied the SSR model to functional time series. Sangalli et al. [31] refined the SSR model and showed that it can be extended to data with three dimensions, including volumes and surfaces that are embedded in three-dimensional spaces. Nguyen et al. [14] implemented this methodology to surface clustering and classification by extending the SSR model to the MSSR$_\mathrm{m}$ model. In this section, we will describe the MSSR$_\mathrm{m}$ model.

2.1. Mixture of spatial spline regression model with mixed effects

Let t be an index for time, at time t, $t = 1 \leq t \leq T$. Denote $\boldsymbol{y}_t$, an $m_t \times 1$ vector, as the observation at time t, where m_t is the length of observation $\boldsymbol{y}_t$. Suppose observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$ can be grouped into K clusters.

Let $\boldsymbol{S}_t$ be the spatial covariates matrix at time t. The matrix $\boldsymbol{S}_t$ has dimension $m_t \times d$ and is calculated from the tent shaped piecewise linear nodal basis functions (NBFs) [11], which will be introduced in Section 2.2, and d is fixed and refers to the number of basis functions. Let $\boldsymbol{\beta}_k$, a $d \times 1$ vector, be the model fixed effects for the kth cluster and π_k be the corresponding cluster allocation probability for observations. We use $\boldsymbol{b}_{tk}$, a $d \times 1$ vector, to denote the random effect coefficients of the kth cluster for $\boldsymbol{y}_t$. We let $\boldsymbol{e}_{tk}$, a $d \times 1$ vector, represent the random error of the kth cluster for observation t. The MSSR$_\mathrm{m}$ model for the observation at time t can be expressed as

$$\boldsymbol{y}_t = \sum_{k=1}^{K} \pi_k (\boldsymbol{S}_t \boldsymbol{\beta}_k + \boldsymbol{S}_t \boldsymbol{b}_{tk} + \boldsymbol{e}_{tk}), \tag{1}$$

where $\boldsymbol{b}_{tk} \sim MVN(0, \xi_k^2 \boldsymbol{I}_d)$ and $\boldsymbol{e}_{tk} \sim MVN(0, \sigma_k^2 \boldsymbol{I}_{m_t})$. Here $MVN(\cdot,\cdot)$ denotes the multivariate normal distribution, and $\boldsymbol{I}_l$ denotes an identity matrix with dimension $l \times l$. Let $^\intercal$ be a symbol denoting the transpose of a vector or matrix. Denote $\boldsymbol{\pi} = (\{\pi_k\}_{k=1}^K)^\intercal$, $\boldsymbol{\beta} = (\{\boldsymbol{\beta}_k\}_{k=1}^K)^\intercal$, $\boldsymbol{\sigma}^2 = (\{\sigma_k^2\}_{k=1}^K)^\intercal$ and $\boldsymbol{\xi}^2 = (\{\xi_k^2\}_{k=1}^K)^\intercal$. Our interest is to conduct Bayesian inference for $\boldsymbol{\theta} = (\boldsymbol{\pi}^\intercal, \boldsymbol{\beta}^\intercal, (\boldsymbol{\sigma}^2)^\intercal, (\boldsymbol{\xi}^2)^\intercal)^\intercal$.

2.2. Linear nodal basis function

In this section, we focus on the nodal basis functions (NBFs) [11]. While B-splines are often used to approximate the univariate functions, NBFS can be used in a similar way to approximate surfaces. As argued by [14], the linear NBFs [11] are useful for problems involving clustering and classification.

The linear NBF is a 'tent shaped' piecewise linear function with shape parameter $\boldsymbol{\delta} = (\delta_1, \delta_2)^\intercal$, centre parameter $\mathbf{c} = (c_1, c_2)^\intercal$ and coordinates x_1, x_2 on a rectangular domain, where δ_1, δ_2 are positive real numbers representing the horizontal shape parameter and vertical shape parameter separately. For any coordinates $\boldsymbol{\eta} = (\eta_1, \eta_2)^\intercal$ on a rectangular domain $R = [\eta_1^-, \eta_1^+] \times [\eta_2^-, \eta_2^+]$, let $\eta_1' = (\eta_1 - c_1)/\delta_1$, $\eta_2' = (\eta_2 - c_2)/\delta_2$, $\boldsymbol{\eta}' = (\eta_1', \eta_2')^\intercal$. The exact form of linear NBF of $\boldsymbol{\eta}$ is defined in Equation (2) [11] as follows:

$$s(\boldsymbol{\eta};\mathbf{c},\boldsymbol{\delta}) = \begin{cases} 1+{\eta_2}' & \text{if } \boldsymbol{\eta}' \in \{\boldsymbol{\eta}' : -1 \le \eta_1' \le 0,\ -1 \le \eta_2' \le \eta_1'\}, \\ 1+{\eta_1}' & \text{if } \boldsymbol{\eta}' \in \{\boldsymbol{\eta}' : -1 \le \eta_1' \le 0,\ \eta_1' \le \eta_2' \le 0\}, \\ 1+{\eta_1}'-{\eta_2}' & \text{if } \boldsymbol{\eta}' \in \{\boldsymbol{\eta}' : -1 \le \eta_1' \le 0,\ 0 \le \eta_2' \le \eta_1'+1\}, \\ 1-{\eta_1}'+{\eta_2}' & \text{if } \boldsymbol{\eta}' \in \{\boldsymbol{\eta}' : 0 \le \eta_1' \le 1, \eta_1'-1 \le \eta_2' \le 0\}, \\ 1-{\eta_1}' & \text{if } \boldsymbol{\eta}' \in \{\boldsymbol{\eta}' : 0 \le \eta_1' \le 1, 0 \le \eta_2' \le \eta_1'\}, \\ 1-{\eta_2}' & \text{if } \boldsymbol{\eta}' \in \{\boldsymbol{\eta}' : 0 \le \eta_1' \le 1, \eta_1' \le \eta_2' \le 1\}, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Similarly to B-splines, the number of nodal basis functions used to approximate surfaces has to be specified. Suppose we use $d = d_1 \times d_2$ basis functions to approximate the surfaces over a fixed rectangular domain. The rectangular domain is divided into $(d_1 - 1) \times (d_2 - 1)$ small grids evenly, $d_1 - 1$ grids in each row and $d_2 - 1$ grids in each column. Nodes on the grids are centres of these nodal basis, and the horizontal length and vertical height on small grids are $\delta_1 = (\eta_1^+ - \eta_1^-)/(d_1 - 1), \delta_2 = (\eta_2^+ - \eta_2^-)(d_2 - 1)$, respectively. The covariates matrix $\boldsymbol{S}_t$ for $\boldsymbol{y}_t$, $1 \le t \le T$, is defined as

$$\boldsymbol{S}_t = [s(\boldsymbol{\eta}_{t,i}; \boldsymbol{c}_j, \boldsymbol{\delta})]_{1 \le i \le m_t, 1 \le j \le d}, \tag{3}$$

where $s(\boldsymbol{\eta}_{t,i}; \boldsymbol{c}_j, \boldsymbol{\delta})$ represents the element in the ith row and jth column of $\boldsymbol{S}_t$, $\boldsymbol{\eta}_{t,i}$, $1 \le i \le m_t$, are coordinates of $\boldsymbol{y}_t$, and $\boldsymbol{c}_j = (c_{j1}, c_{j2})^\top$, $j = 1, \ldots, d$ are the centres and can be obtained by setting $\mathbf{c}_1 = (\eta_1^-, \eta_2^-)^\top, \mathbf{c}_2 = (\eta_1^- + \delta_1, \eta_2^-)^\top, \ldots, \mathbf{c}_{d-1} = (\eta_1^+, \eta_2^- + (d_2 - 1)\delta_2)^\top$, $\mathbf{c}_d = (\eta_1^+, \eta_2^+)^\top$. For example, given a fixed rectangular domain $R = [-1, 1] \times [-1, 1]$, suppose we set the number of NBFs $d = 3 \times 3$, therefore $\delta_1 = \delta_2 = 1$, and the corresponding 9 NBFs centres are $\mathbf{c}_1 = (-1, -1)^\top, \mathbf{c}_2 = (-1, 0)^\top, \mathbf{c}_3 = (-1, 1)^\top, \mathbf{c}_4 = (0, -1)^\top, \mathbf{c}_5 = (0, 0)^\top, \mathbf{c}_6 = (0, 1)^\top, \mathbf{c}_7 = (1, -1)^\top, \mathbf{c}_8 = (1, 0)^\top$ and $\mathbf{c}_9 = (1, 1)^\top$.

3. Model inference

In this section, we introduce two methodologies to make inference on the mixture of spatial spline regression model. Before introducing the inference methods, we discuss the Bayesian framework for this MSSR_m model. We first incorporate the auxiliary variable z_t $(t = 1, \ldots, T)$ in the mixture model to indicate the cluster label of observation $\boldsymbol{y}_t$. Denote $\boldsymbol{Y} = (\{\boldsymbol{y}_t^\top\}_{t=1}^T)^\top, \boldsymbol{Z} = (\{z_t\}_{t=1}^T)^\top, \boldsymbol{b} = (\{\boldsymbol{b}_t^\top\}_{t=1}^T)^\top, \boldsymbol{b}_t = \{\boldsymbol{b}_{t1}^\top, \ldots, \boldsymbol{b}_{tK}^\top\}^\top$. Recall that $\boldsymbol{\theta} = (\boldsymbol{\pi}^\top, \boldsymbol{\beta}^\top, (\boldsymbol{\sigma}^2)^\top, (\boldsymbol{\xi}^2)^\top)^\top$, conditional on z_t, $\boldsymbol{b}_t$ and $\boldsymbol{\theta}$, we have

$$f(\boldsymbol{y}_t | z_t, \boldsymbol{b}_t, \boldsymbol{\theta}) = \prod_{k=1}^{K} \left\{\phi(\boldsymbol{y}_t; \boldsymbol{S}_t\boldsymbol{\beta}_k + \boldsymbol{S}_t\boldsymbol{b}_{tk}, \sigma_k^2 \boldsymbol{I}_{m_t})\right\}^{\mathbf{1}_k(z_t)}, \tag{4}$$

where $\phi(\cdot; \boldsymbol{S}_t\boldsymbol{\beta}_k + \boldsymbol{S}_t\boldsymbol{b}_{tk}, \sigma_k^2 \boldsymbol{I}_{m_t})$ denotes the density of a multivariate normal distribution with mean $\boldsymbol{S}_t\boldsymbol{\beta}_k + \boldsymbol{S}_t\boldsymbol{b}_{tk}$, and covariance $\sigma_k^2 \boldsymbol{I}_{m_t}$. Here $\mathbf{1}(\cdot)$ is an indicator function: $\mathbf{1}_k(z_t) = 1$ if $z_t = k$; otherwise $\mathbf{1}_k(z_t) = 0$.

The joint density (likelihood) of $\boldsymbol{Y}$ conditional on $\boldsymbol{Z}, \boldsymbol{b}, \boldsymbol{\theta}$ can be written as

$$f(\boldsymbol{Y} | \boldsymbol{Z}, \boldsymbol{b}, \boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\xi}^2, \boldsymbol{\sigma}^2) = \prod_{t=1}^{T} \prod_{k=1}^{K} \left\{\phi(\boldsymbol{y}_t; \boldsymbol{S}_t\boldsymbol{\beta}_k + \boldsymbol{S}_t\boldsymbol{b}_{tk}, \sigma_k^2 \boldsymbol{I}_{m_t})\right\}^{\mathbf{1}_k(z_t)}. \tag{5}$$

3.1. Bayesian framework

In this section, we build a hierarchical MSSR$_m$ model under the Bayesian framework. Let a Dirichlet distribution with hyperparameters $(\alpha_1, \ldots, \alpha_K)$ be the prior for the allocation parameter $\boldsymbol{\pi}$, a multinomial distribution with parameter $\boldsymbol{\pi}$ be the prior for the label z_t, $1 \leq t \leq T$, a multivariate normal distribution with hyperparameters $\boldsymbol{\mu}_0$, $\boldsymbol{\Sigma}_0$ be the prior of the fixed-effects coefficients. We use an inverse-gamma distribution with a shape parameter a_0 (or g_0) and a scale parameter b_0 (or h_0) as the prior for the variance parameter ξ_k^2 (or σ_k^2) for $1 \leq k \leq K$ and use a multivariate normal distribution with parameters ξ_k^2 to model the random-effects coefficients $\boldsymbol{b}_{tk}$ for $1 \leq k \leq K, 1 \leq t \leq T$. Hobert and Casella [32] showed that improper selection of prior distributions for mixed model may lead to an improper posterior, while inference method may not give warning (e.g. Gibbs chain). Hence, it is important to check that the posterior distribution is proper. Hobert and Casella [32] provide theorems and numerical examples for this issue, and we refer readers to their work for a more detailed discussion.

The hierarchical MSSR$_\mathrm{m}$ model can be expressed as follows:

$$\boldsymbol{y}_t | z_t = k, \boldsymbol{\beta}_k, \boldsymbol{b}_{tk}, \sigma_k^2 \sim MVN(\boldsymbol{S}_t\boldsymbol{\beta}_k + \boldsymbol{S}_t\boldsymbol{b}_{tk}, \sigma_k^2 \boldsymbol{I}_{m_t}), \tag{6}$$

$$\boldsymbol{\pi} \sim Dir(\alpha_1, \ldots, \alpha_K), \tag{7}$$

$$z_t \sim Mult(\pi_1, \ldots, \pi_K), \tag{8}$$

$$\boldsymbol{\beta}_k \sim MVN(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0), \tag{9}$$

$$\boldsymbol{b}_{tk} | \xi_k^2 \sim MVN(0_d, \xi_k^2 \boldsymbol{I}_d), \tag{10}$$

$$\xi_k^2 \sim IG(a_0, b_0), \tag{11}$$

$$\sigma_k^2 \sim IG(g_0, h_0), \tag{12}$$

where $1 \leq k \leq K$, $1 \leq t \leq T$, $a_0, b_0, g_0, h_0, \alpha_1, \ldots, \alpha_K, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0$ are hyperparameters. Therefore, the normalized posterior distribution for $(\boldsymbol{\theta}^\top, \boldsymbol{b}^\top, \boldsymbol{Z}^\top)^\top$ can be written as

$$\begin{aligned} p(\boldsymbol{\theta}, \boldsymbol{b}, \boldsymbol{Z} | \boldsymbol{Y}) = {} & \frac{1}{f(\mathbf{Y})} \times f(\boldsymbol{\pi}) \prod_{k=1}^{K} f(\boldsymbol{\beta}_k) f(\xi_k^2) f(\sigma_k^2) \\ & \times \prod_{t=1}^{T} \prod_{k=1}^{K} f(\boldsymbol{b}_{tk} | \xi_k^2) \left\{ \phi(\boldsymbol{y}_t | \boldsymbol{S}_t \boldsymbol{\beta}_k + \boldsymbol{S}_t \boldsymbol{b}_{tk}, \sigma_k^2 \boldsymbol{I}_{m_t}) \pi_k \right\}^{\mathbf{1}_k(z_t)}, \end{aligned}$$

where $f(\cdot)$ refers generically to the density of its argument, and $f(\boldsymbol{Y})$ refers to the marginal likelihood and can be evaluated by

$$\begin{aligned} f(\mathbf{Y}) = \int \cdots \int f(\boldsymbol{\pi}) \prod_{k=1}^{K} f(\boldsymbol{\beta}_k) f(\xi_k^2) f(\sigma_k^2) \prod_{t=1}^{T} \prod_{k=1}^{K} \Bigg\{ & f(\boldsymbol{b}_{tk} | \xi_k^2) \\ & \times \left\{ \phi(\boldsymbol{y}_t | \boldsymbol{S}_t \boldsymbol{\beta}_k + \boldsymbol{S}_t \boldsymbol{b}_{tk}, \sigma_k^2 \boldsymbol{I}_{m_t}) \pi_k \right\}^{\mathbf{1}_k(z_t)} \Bigg\} \, \mathrm{d}\boldsymbol{\theta} \, \mathrm{d}\boldsymbol{b} \, \mathrm{d}Z. \end{aligned} \tag{13}$$

However, the integral in Equation (13) is intractable since it involves integrating over all possible values of $\boldsymbol{Z}, \boldsymbol{b}, \boldsymbol{\theta}$. We propose two methods to estimate the normalized posterior distribution for $\boldsymbol{\theta}$, which will be described in the next two sections.

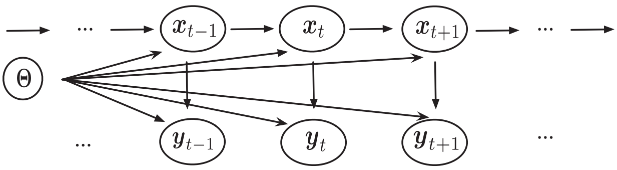

Figure 1. Graphical representation of a simple state space model.

Under the above Bayesian framework, we have

$$f(\boldsymbol{y}_t, \boldsymbol{b}_t, z_t|\boldsymbol{\theta}) = \prod_{k=1}^{K} f(\boldsymbol{b}_{tk}|\xi_k^2)\left\{\phi(\boldsymbol{y}_t|\boldsymbol{S}_t\boldsymbol{\beta}_k + \boldsymbol{S}_t\boldsymbol{b}_{tk}, \sigma_k^2\boldsymbol{I}_{m_t})\pi_k\right\}^{\boldsymbol{1}_k(z_t)}. \tag{14}$$

Taking integration of Equation (14) over $\boldsymbol{b}_t$, we have

$$f(\boldsymbol{y}_t, z_t|\boldsymbol{\theta}) = \prod_{k=1}^{K}\{\phi(\boldsymbol{y}_t; \boldsymbol{S}_t\boldsymbol{\beta}_k, \xi_k^2\boldsymbol{S}_t\boldsymbol{S}_t^{\mathsf{T}} + \sigma_k^2\boldsymbol{I}_{m_t})\pi_k\}^{\boldsymbol{1}_k(z_t)}. \tag{15}$$

Taking integration of Equation (15) over z_t, we have

$$f(\boldsymbol{y}_t|\boldsymbol{\theta}) = \sum_{k=1}^{K}\pi_k\phi(\boldsymbol{y}_t; \boldsymbol{S}_t\boldsymbol{\beta}_k, \xi_k^2\boldsymbol{S}_t\boldsymbol{S}_t^{\mathsf{T}} + \sigma_k^2\boldsymbol{I}_{m_t}). \tag{16}$$

3.2. Markov chain Monte Carlo

Markov chain Monte Carlo is the most commonly used approach for the implementation of Bayesian inference. The basic idea is to construct an ergodic irreducible Markov chain which admits the normalized posterior as its stationary and limiting distribution. The algorithm is run sufficiently long for a burn-in period so that subsequent draws after this period are approximate draws from the posterior. The availability of the conditional posterior distribution for all parameters of interest allows us to estimate the mixture of spatial spline regression in the Gibbs sampling framework. See Algorithm 3 in Appendix 2 for the MCMC inference of the hierarchical MSSR$_{\mathrm{m}}$ model. Also the full conditional distributions are derived in Appendix 2.

3.3. Online sequential Monte Carlo with Gibbs moves

Sequential Monte Carlo methods [22,23] were developed to analyse dynamical models. The most popular application is in state space models. In a state space model, which is graphically displayed in Figure 1, we let $\boldsymbol{x}_t$ denote the latent variable at time t, and assume it is specified by $f_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, where $\boldsymbol{\theta}$ refers to the model static parameter. Let $\boldsymbol{y}_t$ denote the observation at time t. We assume that the observations $\boldsymbol{y}_t$'s are independent conditional on $\boldsymbol{x}_t$'s, and the distribution of $\boldsymbol{y}_t$ is specified by $f_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{x}_t)$. For simplicity, we adopt the notation $\boldsymbol{a}_{1:t}$ to be an abbreviation of $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_t\}$. The target distribution is $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T}) \propto \prod_{t=1}^{T} f_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})f_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{x}_t)$, where $f_{\boldsymbol{\theta}}(\boldsymbol{x}_1|\boldsymbol{x}_0) = f_{\boldsymbol{\theta}}(\boldsymbol{x}_1)$.

To approximate $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T})$ using a standard SMC algorithm described in Algorithm 1, we introduce a sequence of intermediate target distributions $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$ $(t = 1, 2, \ldots, T)$. Each intermediate target distribution is approximated by a set of weighted samples, also called particles in the SMC literature. More specifically, at time t, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$ is approximated by $\{\boldsymbol{x}_{1:t}^{(i)}, W_t^{(i)}\}_{1\leq i\leq N}$. In our notation, superscripts and subscripts respectively refer to the particle and time indices.

The standard SMC algorithm iterates between the following three steps to approximate the intermediate targets: resampling, propagating and weighting. At each iteration t, we first conduct a resampling step to prune particles at iteration $t-1$ with small weights. The path degeneracy issue is well known in SMC literature [33,34,53]. The earlier approximation of intermediate target distributions may collapse as t increases. The purpose of resampling step is to alleviate the path degeneracy issue. A commonly used resampling algorithm is multinomial resampling. Equally weighted particles are produced after multinomial resampling. Then we propose new particles from a proposal distribution. Finally, we compute the weights for each proposed particle.

In Algorithm 1, $q_{t,\boldsymbol{\theta}}(\cdot)$ denotes the proposal distribution for $\boldsymbol{x}_t$, $w_t^{(i)}$ refers to the unnormalized weight and we use $W_t^{(i)}$ to represent the normalized version. If we choose the proposal distribution for $\boldsymbol{x}_t^{(i)}$ to be the prior $f_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(A_t^i)})$, we can obtain a simplified weight update form of $w_t^{(i)} = f_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)})$. However, $f_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(A_t^i)})$ does not take advantage of any

Algorithm 1 Standard sequential Monte Carlo for a simple state space model.

```
Input: data $\boldsymbol{y}_{1:T}$, parameter $\boldsymbol{\theta}$.
Output: $\{\boldsymbol{x}_{1:T}^{(i)}, W_T^{(i)}\}_{1\leq i\leq N}$.
for $i = 1$ to $N$ do
    Draw $\boldsymbol{x}_1^{(i)} \sim q_{1,\boldsymbol{\theta}}(\cdot|\boldsymbol{y}_1)$.
    Update weights $W_1^{(i)} = w_1^{(i)} / \sum_{i=1}^N w_1^{(i)}$, where
        $w_1^{(i)} = \frac{f_{\boldsymbol{\theta}}(\boldsymbol{y}_1|\boldsymbol{x}_1^{(i)}) f_{\boldsymbol{\theta}}(\boldsymbol{x}_1^{(i)})}{q_{1,\boldsymbol{\theta}}(\boldsymbol{x}_1^{(i)}|\boldsymbol{y}_1)}.$    (17)
end for
for $t = 2$ to $T$ do
    for $i = 1$ to $N$ do
        Resample the ancestor index $A_t^i \sim Mult(\{W_{t-1,\boldsymbol{\theta}}^{(j)}\}_{1\leq j\leq N})$.
        Sample $\boldsymbol{x}_t^{(i)} \sim q_{t,\boldsymbol{\theta}}(\cdot|\boldsymbol{x}_{t-1}^{(A_t^i)}, \boldsymbol{y}_t)$.
        Update weights $W_t^{(i)} = w_t^{(i)} / \sum_{i=1}^N w_t^{(i)}$, where
            $w_t^{(i)} = \frac{f_{\boldsymbol{\theta}}(\boldsymbol{y}_t|x_t^{(i)}) f_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(A_t^i)})}{q_{t,\boldsymbol{\theta}}(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(A_t^i)}, \boldsymbol{y}_t)}.$    (18)
    end for
end for
```

Table 1. Notations of a general state space model and a hierarchical MMSR$_\text{m}$ model.

Notations	Simple state space model	Hierarchical MSSR$_\text{m}$ model
Observation	$\boldsymbol{y}_t$	$\boldsymbol{y}_t$
Latent variable	$\boldsymbol{x}_t$	$\boldsymbol{x}_t = (z_t, \boldsymbol{b}_t^\mathsf{T})^\mathsf{T}, \boldsymbol{b}_t = \{\boldsymbol{b}_{t1}^\mathsf{T}, \dots, \boldsymbol{b}_{tK}^\mathsf{T}\}^\mathsf{T}$
Static Parameter	$\boldsymbol{\theta}$ (known)	$\boldsymbol{\theta} = (\boldsymbol{\pi}^\mathsf{T}, \boldsymbol{\beta}^\mathsf{T}, (\boldsymbol{\sigma}^2)^\mathsf{T}, (\boldsymbol{\xi}^2)^\mathsf{T})^\mathsf{T}$ (unknown)
Prior	$f_{\boldsymbol{\theta}}(\boldsymbol{x}_t\|\boldsymbol{x}_{t-1})$	$f_{\boldsymbol{\theta}}(\boldsymbol{x}_t\|\boldsymbol{x}_{t-1}) = f_{\boldsymbol{\theta}}(\boldsymbol{x}_t)$
Proposal	$p_{\boldsymbol{\theta}}(\boldsymbol{x}_t\|\boldsymbol{x}_{t-1}, \boldsymbol{y}_t)$	$p_{\boldsymbol{\theta}}(\boldsymbol{x}_t\|\boldsymbol{x}_{t-1}, \boldsymbol{y}_t) = p_{\boldsymbol{\theta}}(\boldsymbol{x}_t\|\boldsymbol{y}_t)$
Weight w_t	$f_{\boldsymbol{\theta}}(\boldsymbol{y}_t\|\boldsymbol{x}_{t-1})$	$f_{\boldsymbol{\theta}}(\boldsymbol{y}_t\|\boldsymbol{x}_{t-1}) = f(\boldsymbol{y}_t\|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) = f(\boldsymbol{y}_t\|\boldsymbol{\theta})$

information carried by the observations. With this choice of simple proposal distribution, the performance of SMC algorithm will be inefficient if the observations are informative. A more efficient importance proposal distribution is the 'partial posterior' distribution [24,27,28] (also known as 'optimal proposal distribution' in the literature), $q_{t,\boldsymbol{\theta}}(\boldsymbol{x}_t) = p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(A_t^i)}, \boldsymbol{y}_t) = f_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(A_t^i)}) f_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{x}_t)/f_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{x}_{t-1}^{(A_t^i)})$, in which the numerator is the same as the numerator in Equation (17) and Equation (18) and therefore it can be cancelled. In this case, the unnormalized weight takes the form $w_t^{(i)} = f_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{x}_{t-1}^{(A_t^i)})$.

Now we extend the above standard SMC scheme to the MSSR$_\text{m}$ model to conduct online inference. All the notations related to the MSSR$_\text{m}$ model in this section are consistent with the notations used before. Recall that z_t is the cluster label (latent variable) in the model and $\boldsymbol{\theta}$ is a vector of unknown static parameters, and $\boldsymbol{b}_t = \{\boldsymbol{b}_{t,1}, \dots, \boldsymbol{b}_{t,K}\}$, where $\boldsymbol{b}_{t,k}$ is the random effect coefficients of cluster k for $\boldsymbol{y}_t$. Let $\boldsymbol{x}_t = (z_t, \boldsymbol{b}_t)$. The hierarchical MMSR$_\text{m}$ model specified in Section 3.1 is also a state space model, with notations listed in Table 1. We choose the 'partial posterior' as proposals for the latent variable, i.e. $q_{t,\boldsymbol{\theta}}(\boldsymbol{x}_t) = p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_t) = p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t) = p_{\boldsymbol{\theta}}(z_t, \boldsymbol{b}_t|\boldsymbol{y}_t) = p(z_t|\boldsymbol{y}_t, \boldsymbol{\theta})p(\boldsymbol{b}_t|z_t, \boldsymbol{y}_t, \boldsymbol{\theta})$, which is a product of $p(z_t|\boldsymbol{\theta}, \boldsymbol{y}_t)$, partial posterior of z_t and $p(\boldsymbol{b}_t|z_t, \boldsymbol{y}_t, \boldsymbol{\theta})$, full conditional distribution of $\boldsymbol{b}_t$. Using $p(z_t|\boldsymbol{\theta}, \boldsymbol{y}_t) \propto f(z_t|\boldsymbol{\theta})f(\boldsymbol{y}_t|z_t, \boldsymbol{\theta})$, $p(z_t|\boldsymbol{\theta}, \boldsymbol{y}_t)$ can be expressed as

$$p(z_t|\boldsymbol{\theta}, \boldsymbol{y}_t) \sim Mult(1; \tau_{t1}^*, \dots, \tau_{tK}^*), \tag{19}$$

where $\tau_{tk}^* \propto \phi(\boldsymbol{y}_t; \boldsymbol{S}_t\boldsymbol{\beta}_{t,k}, \xi_{t,k}^2\boldsymbol{S}_t\boldsymbol{S}_t^T + \sigma_{t,k}^2\boldsymbol{I}_{m_t})\pi_{t,k}$ and $\sum_{k=1}^K \tau_{tk}^* = 1$.

Our objective is to sequentially approximate $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}) \propto f_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:T})f(\boldsymbol{\theta})$, where $f(\boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\theta}$: $f(\boldsymbol{\theta}) = f(\boldsymbol{\pi})\prod_{k=1}^K f(\boldsymbol{\beta}_k)f(\xi_k^2)f(\sigma_k^2)$. Direct estimation of the posterior distribution of $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ is complicated. In order to approximate $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$, we combine the latent variable $\boldsymbol{x}_{1:T}$ with the model static parameters $\boldsymbol{\theta}$ to conduct model inference. We aim to estimate the posterior distribution $p(\boldsymbol{x}_{1:T}, \boldsymbol{\theta}|\boldsymbol{y}_{1:T}) = p(z_{1:T}, \boldsymbol{b}_{1:T}, \boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ in the SMC framework.

Conditional on $(\boldsymbol{y}_{1:t}, z_{1:t}, \boldsymbol{b}_{1:t})$, for $t = 1 \le t \le T$, sufficient statistics $\boldsymbol{s_t}$ for the parameters $\boldsymbol{\theta}$ in MSSR$_\text{m}$ is given in Proposition 3.1.

Proposition 3.1: *The sufficient statistics $\boldsymbol{s}_t$ for the parameters $\boldsymbol{\theta}$ in MSSR$_\text{m}$ model given $(\boldsymbol{y}_{1:t}, z_{1:t}, \boldsymbol{b}_{1:t})$ for $1 \le t \le T$ can be written as*

$$\boldsymbol{s}_t = \left\{\sum_{t'=1}^t \boldsymbol{b}_{t'k}^\mathsf{T}\boldsymbol{b}_{t'k}, \sum_{t'=1}^t \mathbf{1}_k(z_{t'}), \sum_{t'=1}^t \mathbf{1}_k(z_{t'})\boldsymbol{S}_{t'}^\mathsf{T}\boldsymbol{y}_{t'}^*, \sum_{t'=1}^t \mathbf{1}_k(z_{t'})\boldsymbol{y}_{t'}^{*\,\mathsf{T}}\boldsymbol{y}_{t'}^*, \sum_{t'=1}^t \mathbf{1}_k(z_{t'})\boldsymbol{S}_{t'}^\mathsf{T}\boldsymbol{S}_{t'}\right\}_{k=1}^K,$$

where $\boldsymbol{y}_{t'}^* = \boldsymbol{y}_{t'} - \boldsymbol{S}_{t'}\boldsymbol{b}_{t'k}$ for $1 \le t' \le t$.

Since $\boldsymbol{s}_t$ is a function of $\boldsymbol{s}_{t-1}$ and $\boldsymbol{y}_t, z_t, \boldsymbol{b}_t$, we denote it as $\boldsymbol{s}_t = T(\boldsymbol{s}_{t-1}, \boldsymbol{y}_t, z_t, \boldsymbol{b}_t)$. Appendix 3 provides the proof of existence of sufficient statistics for the model static parameters.

Similar to the general particle learning (i.e. SMC) procedures of state space models in [27], when model static parameter $\boldsymbol{\theta}$ is unknown and its sufficient statistics exists, the online SMC learning of the hierarchical MSSRm model iterates between the following 4 steps at time t: *Resample* ancestor index, *Propagate* latent variable, *Update sufficient statistics*, update model static parameters $\boldsymbol{\theta}$ using one *Gibbs Move*:

Step 1. Resample Sample ancestor index A_t^i, $i = 1, \ldots, N$, from a multinomial distribution with event probabilities $\{W_{t-1}^{(j)}\}_{1 \le j \le N}$, $W_{t-1}^j \propto f(\boldsymbol{y}_t | \boldsymbol{\theta}_{t-1}^{(j)})$ according to Equation (16). We introduce another set of notations $\{\boldsymbol{s}_{t-1}^{(A_t^i)}, \boldsymbol{\theta}_{t-1}^{(A_t^i)}\}_{1 \le i \le N}$ to represent particles after resampling.

Step 2. Propagate latent variable Sample $z_t{}^{(i)}$ from $p(z_t{}^{(i)} | \boldsymbol{\theta}_{t-1}^{(A_t^i)}, \boldsymbol{y}_t)$ according to Equation (19). Then sample $\boldsymbol{b}_t^i$ according to Equation (A5) in Appendix with $(\boldsymbol{\theta}_t^{(A_t^i)}, z_t^{(i)})$.

Step 3. Update sufficient statistics Update sufficient statistics $\boldsymbol{s}_t^{(i)}$ by letting $\boldsymbol{s}_t^{(i)} = T(\boldsymbol{s}_{t-1}^{(A_t^i)}, \boldsymbol{y}_t, z_t^{(i)}, \boldsymbol{b}_t^{(i)})$ according to **Proposition 3.1**.

Step 4. Gibbs move Update $\boldsymbol{\theta}_t^{(i)}$ with one Gibbs move. Details are shown in Algorithm 2.

Algorithm 2 provides a detailed description for our proposed online SMC algorithm. Once a new observation, for example a new image, arrives, we iterate between the above 'Resample-Propagate-Update' steps with one Gibbs Move to update the parameters. After obtaining the particles $\{z_{1:T}^{(i)}, \boldsymbol{b}_{1:T}^{(i)}, \boldsymbol{\theta}_T^{(i)}\}_{i=1}^N$, we get the approximated marginal posterior density $\hat{p}(\boldsymbol{\theta}|y_{1:T}) = \sum_{i=1}^N \mathbf{1}_{\boldsymbol{\theta}_T^{(i)}}(\boldsymbol{\theta})/N$ by dropping particles $z_{1:T}^{(i)}$ and $\boldsymbol{b}_{1:T}^{(i)}$, where $\mathbf{1}_{\boldsymbol{\theta}_T^{(i)}}(\boldsymbol{\theta}) = 1$ if $\boldsymbol{\theta} = \boldsymbol{\theta}_T^{(i)}$, otherwise $\mathbf{1}_{\boldsymbol{\theta}_T^{(i)}}(\boldsymbol{\theta}) = 0$.

As specified in Algorithm 2, we do not run MCMC within SMC to update the model static parameters $\boldsymbol{\theta}_t^{(i)}$ until $t = n.min$. The purpose is to make sure our algorithm is numerically well behaved as MCMC may be degenerate for models with a very small number of observations. In our numerical experiments, we set *n.min* to a small number (e.g. a value falls between 20 and 100), and we observe this is sufficient. As the size of first batch of input for the proposed online SMC algorithm reaches *n.min*, we start to update the static parameters when new observations arrive.

As argued by [29] and [30], including MCMC moves within SMC to mutate particles can alleviate the progressive degeneration. With the existence of model sufficient statistics, this online SMC learning algorithm will be computationally efficient. Instead of using all of the data in the MCMC move, only sufficient statistics are required to summarize the data.

Recall that K is the total number of clusters, and N^* is the total number of iterations in the MCMC algorithm, as described in Algorithm 3. The MCMC algorithm would take $O(tKN^*)$-time to update model parameters when a new observation $\boldsymbol{y}_t$ arrives, because we have to re-run the algorithm in order to use all data information $\boldsymbol{y}_{1:t}$. The cost of MCMC algorithm is a function of the total number of observations t, total number of MCMC iterations N^* and total number of clusters K. In contrast, the computation of the proposed online SMC algorithm Algorithm 2 at time t takes $O(NK)$-time, which does not depend on the number of observations t with the adoption of sufficient statistics in the algorithm.

The cost only depends on the total number of clusters K and total number of particles N. Consequently, the proposed online SMC algorithm is computationally more efficient than the MCMC algorithm, especially when t is large.

Algorithm 2 Online learning for mixtures spatial spline regression model.

```
Input: data $\boldsymbol{y}_{1:T}$, number of clusters $K$, initial value $\{\boldsymbol{\theta}_0\}$.
Output: $\{\boldsymbol{\theta}_{1:T}^{(i)\mathsf{T}}, \boldsymbol{s}_{1:T}^{(i)\mathsf{T}}, W_T^{(i)\mathsf{T}}\}_{1\le i\le N}^{\mathsf{T}}$.
if $t = 1$ then
    for $i = 1$ to $N$ do
        Draw $z_1^{(i)} \sim p(\cdot|\boldsymbol{y}_1, \boldsymbol{\theta}_0)$ according to Equation (19).
        for $k = 1$ to $K$ do
            Draw $\boldsymbol{b}_{1k}^{(i)} \sim p(\cdot|\boldsymbol{y}_1, \boldsymbol{\theta}_0, z_1^{(i)})$ according to Equation (A5) in Appendix.
        end for
        Update the model sufficient statistics $\boldsymbol{s}_1^{(i)}$ according to Proposition 3.1.
    end for
end if
if $t > 1$ then
    for $i = 1$ to $N$ do
        Compute weights $\{W_{t-1}^{(j)}\}_{j=1}^N$, $W_{t-1}^{(j)} \propto f(\boldsymbol{y}_t|\boldsymbol{\theta}_{t-1}^{(j)})$ according to Equation (16).
        Sample the ancestor index of particle $i$ at time $t$, $A_t^i \sim Mult(\{W_{t-1}^{(j)}\}_{1\le j\le N})$.
        Sample $z_t^{(i)} \sim p_t(\cdot|\boldsymbol{y}_t, \boldsymbol{\theta}_{t-1}^{(A_t^i)})$ according to Equation (19).
        for $k = 1$ to $K$ do
            Sample $\boldsymbol{b}_{tk}^{(i)} \sim p(\cdot|\boldsymbol{y}_t, \boldsymbol{\theta}_{t-1}^{(A_t^i)}, z_t^{(i)})$ according to Equation (A5) in Appendix.
        end for
        Update the model sufficient statistics $\boldsymbol{s}_t^{(i)}$ conditional on $\boldsymbol{s}_{t-1}^{(A_t^i)}$ and $\boldsymbol{y}_t, \boldsymbol{b}_{tk}^{(i)}, z_t^{(i)}$ via
        Proposition 3.1.
        if $t > n.min$ then
            Sample $\boldsymbol{\pi}_t^{(i)} \sim p(\cdot|\boldsymbol{y}_{1:t}, z_{1:t}^{(i)}, \boldsymbol{\beta}_{t-1}^{(A_t^i)}, \boldsymbol{\sigma}^{2(A_t^i)}_{t-1}, \boldsymbol{\xi}^{2(A_t^i)}_{t-1}, \boldsymbol{b}_t^{(i)}, \boldsymbol{s}_t^{(i)})$ according to
            Equation (A3) in Appendix.
            for $k = 1$ to $K$ do
                Sample $\boldsymbol{\beta}_{t,k}^{(i)} \sim p(\cdot|\boldsymbol{y}_{1:t}, z_{1:t}^{(i)}, \boldsymbol{\pi}_t^{(i)}, \sigma^{2(A_t^i)}_{t-1,k}, \xi^{2(A_t^i)}_{t-1,k}, \boldsymbol{b}_{t,k}^{(i)}, \boldsymbol{s}_t^i)$ according to
                Equation (A4) in Appendix.
                Sample $\sigma^{2\ (i)}_{t,k} \sim p(\cdot|\boldsymbol{y}_{1:t}, z_{1:t}^{(i)}, \boldsymbol{\pi}_t^{(i)}, \boldsymbol{\beta}_{t,k}^{(i)}, \xi^{2(A_t^i)}_{t-1,k}, \boldsymbol{b}_{t,k}^{(i)}, \boldsymbol{s}_t^i)$ according to
                Equation (A6) in Appendix.
                Sample $\xi^{2\ (i)}_{t,k} \sim p(\cdot|\boldsymbol{y}_{1:t}, z_{1:t}^{(i)}, \boldsymbol{\pi}_t^{(i)}, \boldsymbol{\beta}_{t,k}^{(i)}, \sigma^{2(i)}_{t,k}, \boldsymbol{b}_{t,k}^{(i)}, \boldsymbol{s}_t^i)$ according to
                Equation (A7) in Appendix.
            end for
        else
            Set $\boldsymbol{\theta}_t^{(i)} = \boldsymbol{\theta}_{t-1}^{(A_t^i)}$.
        end if
    end for
end if
```

3.4. Label switching

There is a parameter nonidentifiability issue in the posterior distribution of mixture model as the likelihood function is identical for the permutation of a part of parameters [35]. The nonidentifiable parameters include the cluster labels. Therefore, this is known as the label switching issue. Due to label switching, the posterior distributions are multimodal with a multiple of $K!$ symmetric modes in case of exchangeable priors. Typically Markov chains would have trouble to visit all those modes in a symmetric manner. Especially, when Gibbs samplers are used for mixture models, label switching often does not occur, which leads to inefficient samplers [36,37]. Although such inefficient samplers often can only obtain samples for one mode of the posterior distribution, they can adequately provide satisfactory parameter estimates in practice.

Our model belongs to Bayesian mixture models and, therefore, has the parameter nonidentifiability issue. Since the proposed online SMC is based on Gibbs moves, label switching may not occur and the samples will be concentrated around one mode of the posterior distribution. However, the marginal likelihood estimate using only samples from one mode would be biased, which will be addressed in Section 4. In case that Markov chains can switch cluster labels and visit multiple modes, the label switching problem can be addressed by posing artificial identifiability constraints [35] and relabeling the mixture components [38].

4. Model selection

Model selection is an important task in the Bayesian MSSR$_\mathrm{m}$ model framework as in many scenarios we are not able to know the optimum model. The goal is to compute the marginal likelihood $p(\boldsymbol{y}_{1:T})$. Numerous methods have been proposed to estimate the marginal likelihood [39–42]. These algorithms require substantial additional effort in both computation and implementation to compute the marginal likelihood, which is not accessible to stream type data.

One advantage of the standard sequential Monte Carlo method is that it can provide an unbiased marginal likelihood estimator as a by-product of the algorithm. This marginal likelihood estimator admits a concise form, which is the product of average unnormalized weights. In the following proposition, we show that the marginal likelihood estimator provided by our proposed algorithm is unbiased, with specific conditions. The proof of this proposition is presented in Appendix 4.

Proposition 4.1: *If we update $\boldsymbol{\theta}_t^{(i)}$ with multiple Gibbs moves until convergence achieved, such that $\boldsymbol{\theta}_t^{(i)} \sim p(\boldsymbol{\theta}|\boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t})$. The product of average unnormalized weights $\prod_{t=1}^{T}\sum_{i=1}^{N} p(\boldsymbol{y}_t|\boldsymbol{\theta}_{t-1}^{(i)})/N$ is an unbiased estimator of the marginal likelihood $p(\boldsymbol{y}_{1:T})$,*

$$E\left(\prod_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{y}_t|\boldsymbol{\theta}_{t-1}^{(i)})\right) = p(\boldsymbol{y}_{1:T}). \tag{20}$$

Practically we only use one Gibbs move to update $\boldsymbol{\theta}_t^{(i)}$, and the marginal likelihood estimator is generally not unbiased as the condition displayed in Proposition 4.1 is broken.

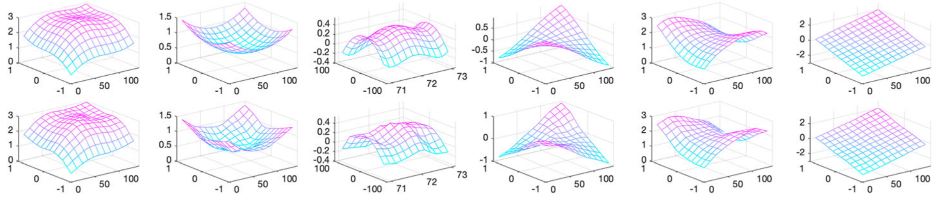

Figure 2. True surfaces versus fitted surfaces: the top row displays the true surfaces of simulated by functions $f_1, \ldots, f_6$ uniformly over the rectangular domain $[-1,1] \times [-1,1]$, and the bottom row is the corresponding MCMC fitted mean surfaces with $d = 6 \times 6$.

In our real data analysis, we compare our model selection criteria with information criterion (*i.e.* Watanabe–Akaike information criteria (WAIC), expected predictive deviance (EPD), expected Akaike information criterion (EAIC) and expected Bayesian information criterion (EBIC)) [41,42] and demonstrate it works in practice.

Due to the label switching issue described in Section 3.4, we hardly obtain samples that can explore the complete posterior distribution for the mixture models, especially for high-dimensional models. When the samples are concentrated around a single mode of the posterior distribution, the estimated log marginal likelihood based on these samples should be corrected by adding $\log(K!)$ [43].

5. Simulation study

We evaluate the proposed algorithms using simulation studies. Assume for each surface cluster k, $k = 1, \ldots, 6$, its common features $f_k(\eta_1, \eta_2)$ over a rectangular domain $[-1,1] \times [-1,1]$ admit the following functional forms:

$$\begin{aligned} f_1(\eta_1,\eta_2) &= \frac{\eta_1^3+\eta_2^3+3}{\sqrt{1+\eta_1^2+\eta_2^2}}, \quad f_2(\eta_1,\eta_2) = \frac{\eta_1^2+\eta_2^2+1}{\sqrt{4+\eta_1^2+\eta_2/4}}, \\ f_3(\eta_1,\eta_2) &= 1-\sin(\eta_1^2+1)+\frac{\cos(1+\eta_2^2)}{2}, \quad f_4(\eta_1,\eta_2) = \sin(\eta_1\eta_2), \\ f_5(\eta_1,\eta_2) &= \cos(\eta_1+\eta_2)+\sin(\eta_1^2)+\cos(\eta_2^2), \quad f_6(\eta_1,\eta_2) = \eta_1+\eta_2. \end{aligned} \tag{21}$$

A surface $\boldsymbol{y}_t$ belonging to cluster k is simulated through

$$\boldsymbol{y}_t = (f_k(\boldsymbol{\eta}_{t,1}), \ldots, f_k(\boldsymbol{\eta}_{t,m_t}))^\top + \boldsymbol{S}_t\boldsymbol{b}_{tk} + \boldsymbol{e}_{tk},$$

where $(f_k(\boldsymbol{\eta}_{t,1}), \ldots, f_k(\boldsymbol{\eta}_{t,m_t}))^\top$ is the common feature for surfaces in cluster k – the mean surface for cluster k, and $\boldsymbol{\eta}_{t,1}, \ldots, \boldsymbol{\eta}_{t,m_t}$ are distributed uniformly over the domain, each surface contains $m_t = 12 \times 12$ points. $\boldsymbol{S}_t$ is the basis covariates function defined in Equation (3). The observed surface is then $\boldsymbol{y}_t = (y_{t,1}, \ldots, y_{t,m_t})$. The random effects and the errors of the model are simulated from $\boldsymbol{b}_{tk} \sim MVN(0, \xi_k^2\boldsymbol{I}_d)$ and $e_{tk} \sim MVN(0, \sigma_k^2\boldsymbol{I}_{m_t})$. The number of nodal basis functions we used in the simulation study is $d = 6 \times 6$. The 6 panels in the first row of Figure 2 display the mean surfaces simulated by functions $(f_k(\boldsymbol{\eta}_{t,1}), \ldots, f_k(\boldsymbol{\eta}_{1,m_t}))^\top$ in Equation (21).

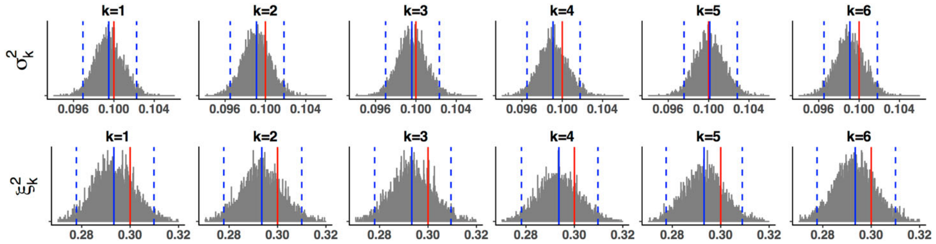

Figure 3. MCMC estimates of σ_k^2 and ξ_k^2, $(k = 1, \ldots, 6)$. The dashed blue lines represent 95% equal tailed credible interval, the solid blue lines indicate the mean value of the estimates and the solid red lines imply the true values of the estimates.

5.1. Surface fitting

In this section, we use a relatively simple experiment to illustrate the $\mathrm{MSSR_m}$ model fitting using the MCMC algorithm. We use Equation (21) to generate the common feature for surfaces in each cluster. A data set comprising 600 images in total (100 for each cluster) is simulated. We set $\sigma_k^2 = 0.1$ and $\xi_k^2 = 0.3$ for $k = 1, \ldots, 6$. The total number of MCMC iterations is set to 5000, and we discard the first 1000 iterations as burn-in. Figure 2 displays a comparison between the true surfaces we simulated (upper panels) and the estimated mean surfaces provided by running the MCMC algorithm (bottom panels). As indicated in Figure 2, the fitted mean surfaces, $\boldsymbol{S}_t\boldsymbol{\beta}_k$, recover the common features of the true surfaces. In Figure 3, we show the histogram of the posterior distributions of σ_k^2 and ξ_k^2 obtained from running MCMC. The red line represents the true value of parameter and the blue line represents the mean value of estimated posterior. The dashed blue lines are the 2.5% and 97.5% quantiles of the posterior distribution. Figure 3 indicates that all of the true values of the parameters can be covered by the associated 95% credible intervals. The estimated posterior means of both σ_k^2 and ξ_k^2 are very close to the corresponding true values.

5.2. Comparison of online SMC with MCMC, EM and DEM

In this section, we simulate two data sets to evaluate the performance of online SMC, MCMC, EM and DEM in terms of sequential image clustering. For MCMC, we use the Gelman–Rubin diagnostic [44] to assess the convergence of the Markov chain. For SMC, we use two ways to check the correctness of our Monte Carlo procedure. The first one is based on the joint distribution testing methodology of [45]; the second one is based on unbiasedness of the marginal likelihood estimate from SMC [46]. We also compare our proposed online SMC to the EM and the Distributed EM (DEM), we refer readers to [16,47] for the details of the EM and DEM, and we set $\gamma = 0.7$ in the DEM as in [16], the convergence is reached when the relative change in the log likelihood between two successive iterations is less than 10^{-4}. All experiments were run on Intel E5-2683 v4 Broadwell @2.1 Ghz machines.

In both data sets, we simulate images in an online fashion where the observation arrives one by one, and the data keep accumulating until the total number of observations reaches $T = 10,000$. The number of particles we set in the online SMC algorithm is $N = 1000$. The

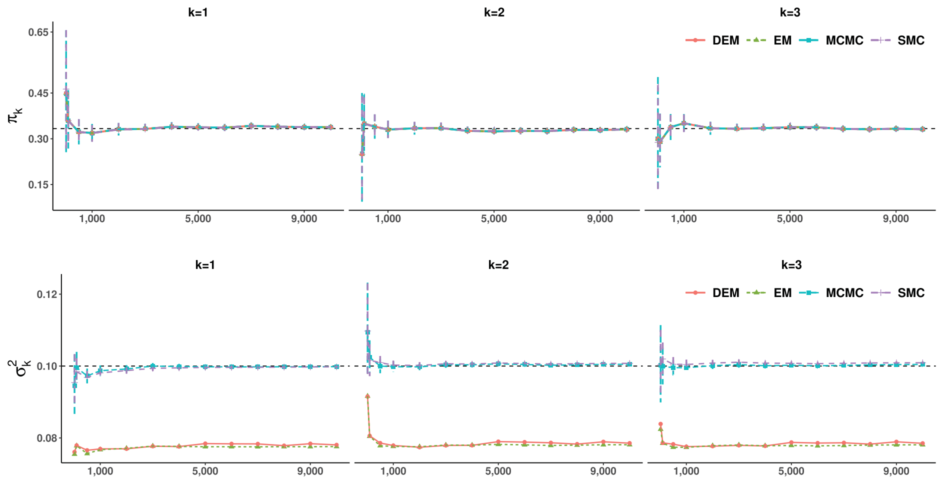

Figure 4. Estimated parameters π_k, σ_k^2 with 95% equal tailed credible interval of online SMC algorithm versus MCMC, EM and DEM, when $K = 3$ and the number of observations increases from $t = 20$ to 10, 000.

number of MCMC iterations is set to 5000 to guarantee the convergence of the algorithm and we burn-in the first 20% of the chain. In our experiments, we observe the samples of both SMC and MCMC are concentrated around a single mode of the posterior distribution.

In the first experiment, we simulate a data set with a relatively small number of clusters, $K = 3$. The common features of images are generated by f_1, f_2 and f_3 of Equation (21). The allocation probability is set to $\pi = (1/3, 1/3, 1/3)$. We assume the variance of random effects and error are $\sigma_k^2 = 0.1$ and $\xi_k^2 = 0.3$ for $k = 1, 2, 3$. In online SMC algorithm, we do not update the model parameters until we have received a small batch of images, here we set $n.min = 20$, a relative small number. The initial value of parameters are obtained by running MCMC on the first 20 images. After that, we update the model parameters once when we obtain a new image. We update the MCMC, EM and DEM with $T = n_{min}, 100, 500, 1000, 2000, \ldots, 10000$ images. Figure 4 displays the comparison of π_k, σ_k^2 as a function of t provided by online SMC, MCMC, EM and DEM. The horizontal lines represent the point estimates (*i.e.* posterior mean provided by MCMC and SMC, MLE provided by EMs) of different methods. The vertical lines represent the 95% credible intervals for online SMC and MCMC. We only display the credible intervals for online SMC at $T = n_{min}, 100, 500, 1000, 2000, \ldots, 10,000$ for the purpose of making a comparison with the MCMC algorithm. As indicated in Figure 4, all algorithms achieve similar performance in terms of estimating parameters π_k $(k = 1, 2, \ldots, K)$, while the EM and the DEM tend to underestimate σ_k^2. With the increment of observations, the posterior mean of both MCMC and online SMC gets closer to the true value, and the credible interval tends to get narrower, as expected.

In our second experiment, we simulate from the MSSR$_\mathrm{m}$ model with more latent states by setting $K = 6$. The common features are simulated from the six functions listed in Equation (21). The allocation probability π_k, random effects ξ_k^2 and variance of the error σ_k^2 are set to $\pi = (1/6, 1/6, 1/6, 1/6, 1/6, 1/6)$, $\sigma_k^2 = 0.1$ and $\xi_k^2 = 0.3$ for

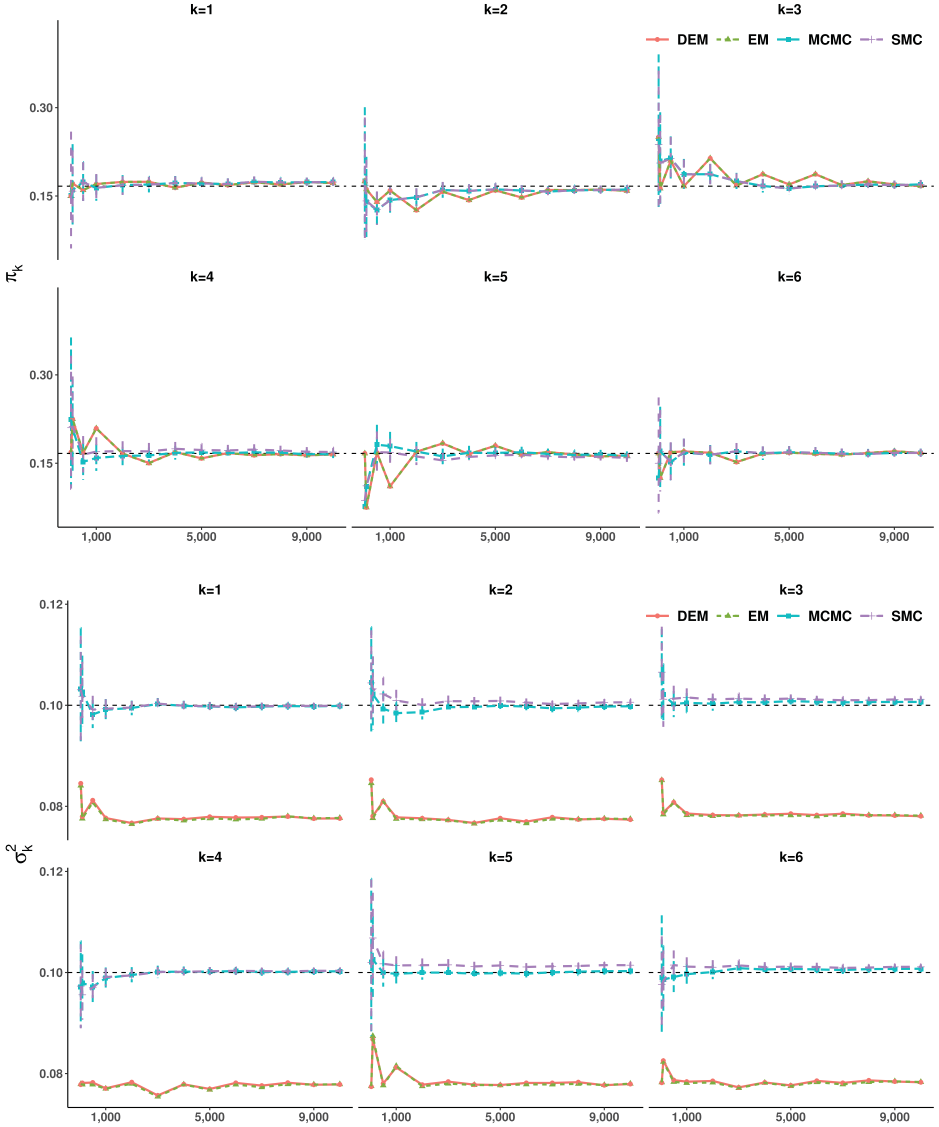

Figure 5. Estimated parameters π_k, σ_k^2 with 95% equal tailed credible interval of online SMC algorithm versus MCMC algorithm (EM, DEM) when $K = 6$ and the number of observations increases from $t = 40$ to $10,000$.

$k = 1, \ldots, 6$. Our online SMC algorithm does not update the parameters until we have $n.\min = 40$ observations. Figure 5 displays the parameter estimates (SMC, MCMC, EM and DEM) and 95% credible intervals (SMC and MCMC) for π_k and σ_k^2 as a function of time t.

All the algorithms achieve similar performance in terms of estimating parameters π_k, for $k = 1, 2, \ldots, K$, while the EM and the DEM underestimate σ_k^2. With the increment of the number of images, the posterior means of π_k and σ_k^2 for both the MCMC and the online SMC tend to converge to the corresponding true value.

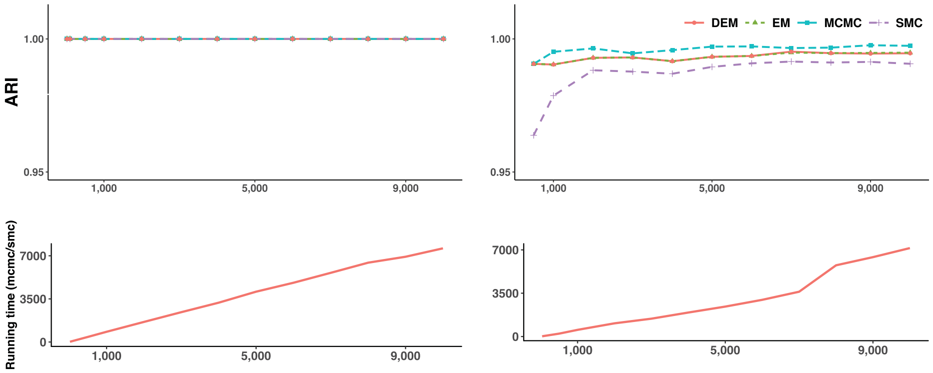

Figure 6. Estimated ARI of online SMC algorithm versus MCMC, EM and DEM $K = 3$ (top left panel) and $K = 6$ (top right panel) and running time ratio of MCMC algorithm over online SMC algorithm when $K = 3$ and (bottom right panel) when $K = 6$ (bottom right panel).

In both experiments, we also apply the Adjusted Rand Index (ARI) [48] to measure the performance of clustering. As indicated in Figure 6, both the online SMC algorithm and the MCMC algorithm can achieve quite a good performance in terms of ARI. However, the running time ratio of MCMC over online SMC increases almost linearly as time evolves, which indicates that online SMC is more scalable to stream type data. For example, as a new observation arrives, it takes less than 10 seconds when $K = 3$ and less than 20 seconds when $K = 6$, to update the model with our proposed online algorithm, while the computational cost of the re-run of the MCMC algorithm is 6.43 hours ($t = 3000,\ K = 3$), 12.77 hours ($t = 6000,\ K = 3$), 7.51 hours ($t = 3000,\ K = 6$) and 15.11 hours ($t = 6000,\ K = 6$). Both experiments indicate that the online SMC algorithm is at least several orders of magnitude faster than the MCMC algorithm in terms of computational cost when t is large (> 2000). Instead of a re-run of the MCMC method, our proposed online SMC algorithm only needs to update the sufficient statistics in **Proposition 3.1** in order to achieve model updating, which leads to efficient computation relative to that of MCMC for sequential image data. With the increment of observations, both algorithms can achieve good performance in terms of parameter estimation and image clustering.

6. Real data analysis

In this section, we apply our proposed online SMC algorithm to two real data sets: one handwritten image data and one brain imaging dataset where brain activity is recorded using MEG. In real applications, one challenge in clustering is the lack of information for the number of clusters. This generally requires some model selection technique to choose the optimum model. As we have alluded in the previous section, the computation of the marginal likelihood $p(y_{1:T})$ in Bayesian statistics is a challenge, but online SMC can provide an unbiased estimator of the marginal likelihood as a by-product of the algorithm, which is a big advantage of SMC over MCMC. Our SMC algorithm is inefficient to sample the full posterior distribution with $K!$ identical modes, and it only explores one mode of the posterior distribution. We adjust the computation of marginal likelihood as described

Table 2. Distribution of Hindu–Arabic handwritten numbers in the sample.

Number	0	1	2	3	4	5	6	7	8	9	Total
Frequency	904	723	500	385	413	344	443	419	408	461	5000

in Section 4. In this section, we investigate the performance of marginal likelihood of online SMC and treat Watanabe–Akaike information criteria (WAIC), expected predictive deviance (EPD), expected Akaike information criterion (EAIC), expected Bayesian information criterion (EBIC) [41,42] as baselines for comparison. The implementation of WAIC, EPD, EAIC and EBIC for our model can be found in Appendix 6.

6.1. Handwritten number images

The first real application we consider is the analysis of handwritten number images. We apply our proposed online SMC algorithm to a subset of the ZIP code data set used in [49]. Every image consists of 16×16 grid of pixels, i.e. each image contains 256 observations and we use 5000 images in total. The images distribution is shown in Table 2.

We set $K = 8, 10, 12$, and for each K, let $d = 6 \times 6, 8 \times 8$ separately. Hence, we have 6 MSSR_m models in total. We mimic a scenario where the images arrive one by one. For the online SMC algorithm, we do not update the model parameters until we have $n.min = 100$ images. After $t = 100$, we update the parameters once one image arrives. The number of particles we use is $N = 1000$. For each pair of K and d, we try several sets of initial values.

The information criterion (*i.e.* WAIC, EPD, EAIC, EBIC) and $\log(p(\boldsymbol{y}_{1:T}))$ provided by online SMC are shown in Table 3. The model with a larger number of basis functions has higher marginalized likelihood and smaller information criterion values, which indicates a larger number of basis function is more preferable. And the performance of the MSSR_m model gets better as K increases as shown in Table 3. The information criterion and $\log(p(\boldsymbol{y}_{1:T}))$ all indicate the MSSR_m model with $K = 12$ and $d = 8 \times 8$ is the optimal model for those considered. However, the information criterion are computationally more expensive, as it takes an extra $O(TN)$-time to compute. We display the common features of the image digits provided by MSSR_m model with $K = 12$ in Figure 7. The MSSR_m model is able to recover all the common features of the 10 digits as well as multiple sub-groups for 0 (the $2nd$, $7th$ clusters), 2 (the $4th$, $9th$, $11th$ clusters), 5 ($5th$, $10th$ clusters). The $1st$ cluster indicates that handwritten digit 9 shares some common features with handwritten digit 7, the $6th$ cluster indicates that handwritten digit 8 shares some common features with 3 and the $9th$ indicates that some of the handwritten digit 8 share same common features as some handwritten digits 2. The common features of models with $K = 8$ and $K = 10$ are displayed in Appendix 5.

6.2. Brain images

In this section, we apply the online SMC algorithm to a subset of images collecting during a neuroimaging study examining the neural response to different natural stimuli. The data are collected using 204 MEG sensors located around the scalp where each sensor measures a time series representing the magnetic field at its location. The magnetic field at a given location is an indirect measurement of the electric neural activity within the brain [50,51].

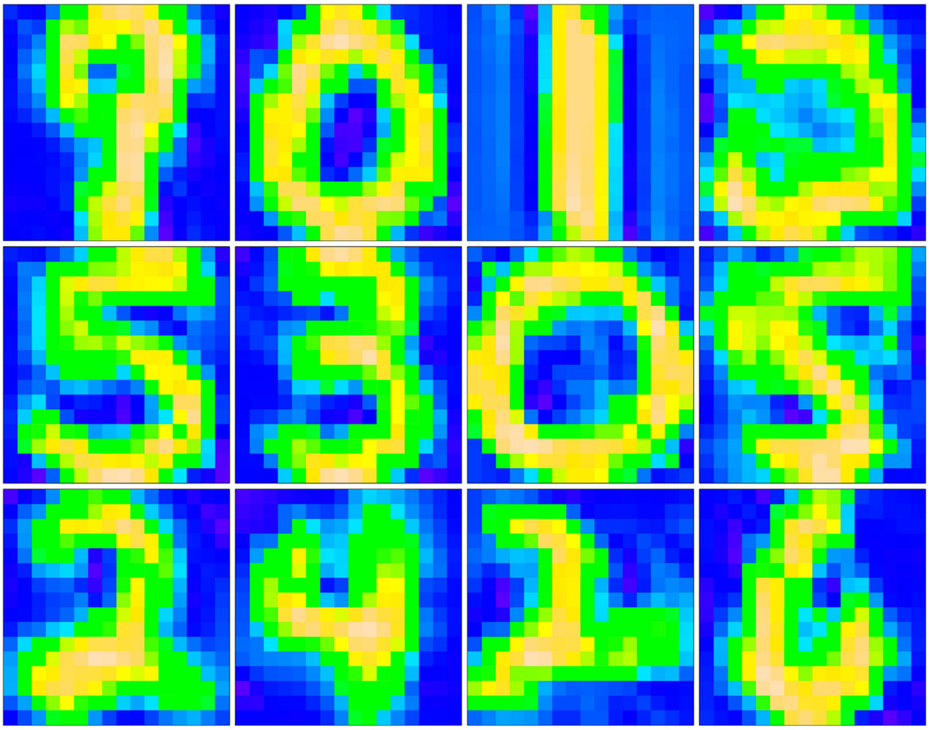

Figure 7. Online SMC estimated common features from MSSR$_\mathrm{m}$ model of handwritten number images by cluster when $K = 12$, $d = 8 \times 8$, labelled as the 1*st* cluster to the 12*th* cluster in left-to-right, top-to-bottom order.

Table 3. Comparison of EPD, EAIC, EBIC and $\log(p(y_{1:T}))$ for MSSR$_\mathrm{m}$ model with $K = 8, 10, 12$ and $d = 6 \times 6, 8 \times 8$ for handwritten number images.

		No. of clusters		
Criteria	d	$K = 8$	$K = 10$	$K = 12$
$\log(p(y_{1:T}))$	6×6	−1452763	−1443349	−1442419
	8×8	−1294875	−1277139	−1274552
EPD	6×6	2901649	2883574	2881265
	8×8	2582130	2548121	2541554
EAIC	6×6	2902273	2884354	2882201
	8×8	2583202	2549461	2543162
EBIC	6×6	2904307	2886895	2885251
	8×8	2586695	2553827	2548402
WAIC	6×6	2901789	2883780	2881520
	8×8	2582337	2548418	2541894

The sensor data at a given time point are projected onto a 2D grid and represented through a 2D image and each recording is made at 200Hz for 1 second resulting in 200 images made in each recording. The study involves over 700 such recordings, where each is associated with one of the five visual stimuli (1. Artificial: screen savers showing animated shapes or text; 2. Nature: clips from nature documentaries, showing natural scenery like mountains or oceans; 3. Football: clips taken from (European) football matches of Spanish La Liga; 4.

Table 4. Distribution of stimuli in the sample data.

Stimulus	1	2	3	4	5
Frequency	1183	1003	706	887	1221

Table 5. Comparison of EPD, EAIC, EBIC and WAIC for MSSR_m model with $K = 5, 7, 9$ and $d = 6 \times 6, 8 \times 8$ for brain images.

		No. of Clusters		
Criteria	d	$K = 5$	$K = 7$	$K = 9$
$\log(p(y_{1:T}))$	6×6	−1549598	−1545574	−1543843
	8×8	−1466517	−1460026	−1456718
EPD	6×6	3098016	3089599	3085886
	8×8	2930890	2917079	2909405
EAIC	6×6	3098406	3090145	3086588
	8×8	2931560	2918017	2910611
EBIC	6×6	3099677	3091924	3088876
	8×8	2933743	2921073	2914541
WAIC	6×6	3098131	3089754	3086072
	8×8	2931129	2917420	2909831

Mr. Bean: clips from the episode Mind the Baby, Mr. Bean of the Mr. Bean television series; 5. Chaplin: clips from the Modern Times feature film, starring Charlie Chaplin).

The neuroimaging data set contains a time series of 200 images for each of 727 recordings. Each time series (sample) is associated with one of the aforementioned five stimuli, that is, what the subject was watching when the time series of images was recorded. Figure 8 shows 2D images of three randomly drawn recordings against the same stimulus at the beginning ($t = 1, 2, 3$), middle ($t = 101, 102, 103$) and end ($t = 198, 199, 200$) of the 1 second recording period. Figure 8 demonstrates the variability in brain activity across different recordings for the same stimulus and thus shows that the type of stimulus associated with a given recording does not clearly distinguish the images from each other, at least by eye. We apply our classifier to decode the images in order to reveal the common stimuli via clustering. We focus our analysis of this application on model selection.

In this application, we use 5000 images, the corresponding stimuli associated with each image is distributed as shown in Table 4. Each image originally consists of 512 × 512 pixels, the high dimensionality of these images makes the implementation challenging. Hence, we begin by removing redundant zeros around the boundaries for each image and compressing the images to a more coarse level, 14 × 18 pixels to save computational cost.

We set $K = 5, 7, 9$, and $d = 6 \times 6, 8 \times 8$ for each K. We assume that the images are received one by one or batch by batch. We do not update the model parameters until we have $n.min = 100$ images. After $t = 100$, we update the parameters once one image arrives. The number of particles we use is $N = 1000$. For each pair of K and d, we try several sets of initial values. The information criterion and $\log(p(y_{1:T}))$ provided by online SMC are shown in Table 5. $\log(p(y_{1:T}))$ and information criterion all indicate the model performs better when more basis functions are used. And the performance of MSSR_m model gets better as K increases as shown in Table 5. They all indicate the MSSR_m model with $K = 9$ and $d = 8 \times 8$ is the optimum.

As indicated in Figure 9, compared to $K = 5$, when $K = 7, 9$, the MSSR_m model is able to capture more common features of several subtle subgroups. As we mentioned in

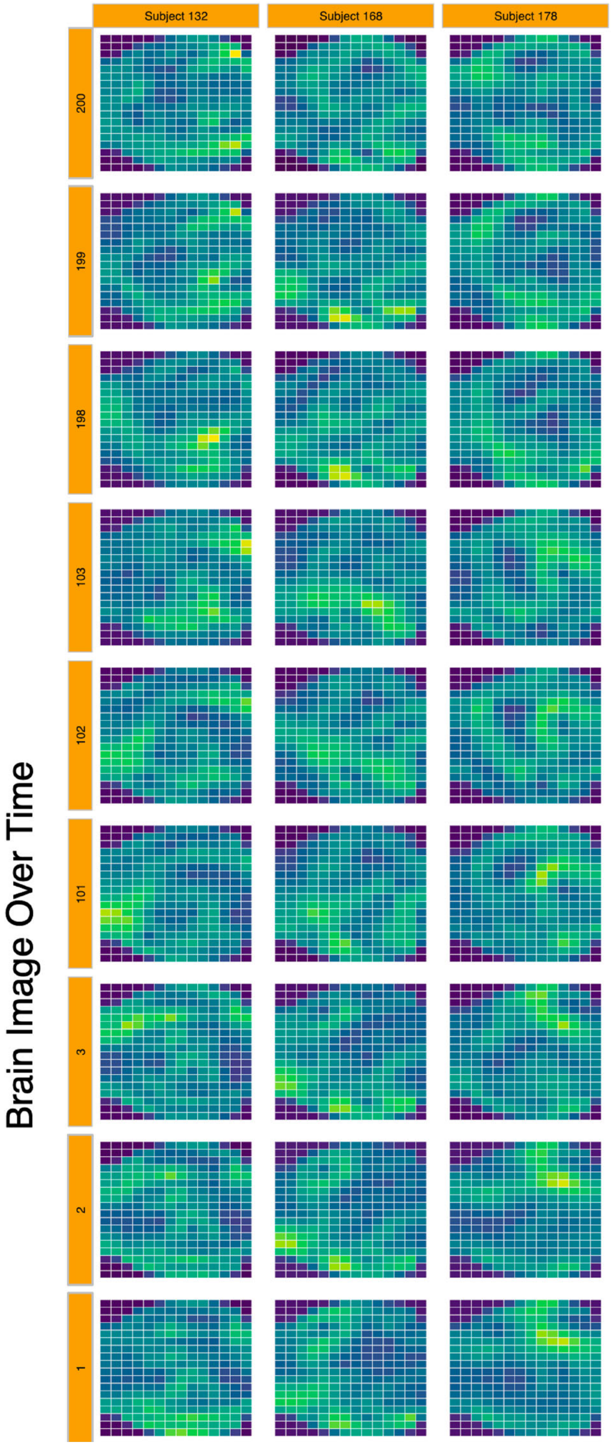

Figure 8. Brain images of three different recordings against same stimulus at the beginning ($t = 1, 2, 3$), middle ($t = 101, 102, 103$) and end ($t = 198, 199, 200$) of the experimental period.

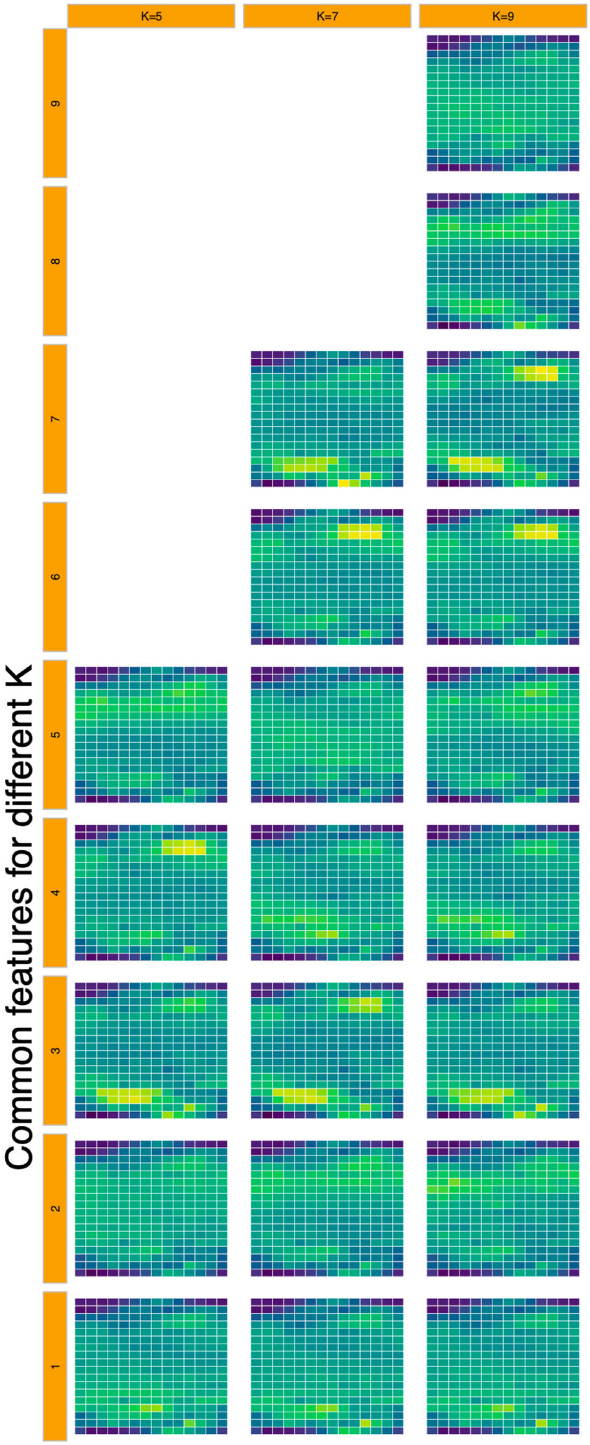

Figure 9. Online SMC estimated common features from MSSR_m model for Brain Images when $K = 5, 7, 9$ and $d = 8 \times 8$.

Section 3.3, the computational cost of updating parameters in the online SMC algorithm is $O(NK)$. Thus the larger is K, the higher is the computational cost. To balance the computational cost and the performance of estimation, in this application of brain images we suggest to take 7 as the appropriate K for the $\mathrm{MSSR_m}$ model, since its estimates result in more subtle discrimination than that of when $K = 5$ and its computational cost is lower than that of when $K = 9$.

7. Discussion

In this article, we derive an MCMC algorithm under the Bayesian framework as an alternative approach to do model inference for the $\mathrm{MSSR_m}$ models. Moreover, we propose an online SMC algorithm to deal with the stream type of image data efficiently via adoption of sufficient statistics and augment variables. When new data arrive, our proposed online SMC algorithm achieves parameter updating in a constant time, which is more adaptable to large sequential image data. In contrast, the required computation time for the MCMC algorithm is a linear function of the total number of observations at time t since it always has to be re-run when new data arrive. Our simulation studies demonstrated that the proposed online SMC algorithm is more efficient than the MCMC algorithm in terms of computing time, while both algorithms can achieve good performance from the perspective of model inference.

Model selection is an important but challenging task in Bayesian statistics. We show that the marginal likelihood estimator provided by our proposed algorithm is unbiased, and it serves as a by-product of the algorithm. We compare this estimator with existing model selection criterion (*e.g.* WAIC, EPD, EAIC, EBIC) and showed that the same models are selected via information criterion and the estimated $\log(p(\boldsymbol{y}_{1:T}))$ from SMC. But information criterion requires extra cost to compute and the computational complexity increases linearly with artificial time T.

In the posterior distribution of MSSR_m models, the likelihood function is identical for all $K!$ permutation of labels. This may induce the label switching and complicate the inference, which makes it difficult to justify the convergence of MCMC [35]. Our developed online SMC algorithm based on importance sampling can circumvent the complicated diagnosis of convergence of Markov chains. On one hand, our current method is inefficient to sample the full posterior distribution with $K!$ modes, which is common for methods based on Gibbs moves for mixture models; on the other hand, it is easier to use the samples that are concentrated on one posterior mode than applying complicated and possibly unsatisfactory strategies to address the label switching issue. Our numerical experiments demonstrate our proposed online algorithm provides an appropriate discrete approximation to the distributions of interest. If we use more efficient proposal distributions rather than the Gibbs moves, we may obtain samples from multiple posterior modes. In that case, we can explore various approaches in the literature that are developed to address the label switching issue [17,35–38,52] of mixture models. But this is out of the scope of this paper.

There are several lines for our future work to improve surface clustering. First of all, one limitation of the current model is that we fix the number of clusters. It is of interest to automatically select the number of clusters K by model and data. One possible way to loosen this constraint is to treat K as a parameter and introduce a Dirichlet process prior for K. Second, the covariance matrix of the random effects is proportional to an identity matrix. A more realistic assumption is to incorporate spatial correlation in the random effects. For example, assuming the random effects arise from a Gaussian process with a suitable spatial covariance. Images for one specific cluster share common features. Another line of future work is to conduct clustering based on the common features by principle component analysis for image data.

Acknowledgments

Research was enabled in part by support provided by WestGrid (www.westgrid.ca) and Compute Canada (www.computecanada.ca). The authors acknowledge support with data storage and computing issues from WestGrid and Compute Canada. The authors thank the editor, associate editor, and referees, for their valuable comments that helped in improving the paper significantly.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research is supported by the Shanghai Science and Technology Program (No. 21010502500), the startup fund of ShanghaiTech University, the Fundamental Research Funds for the Central Universities, Nankai University (No. 63211089), the National Natural Science Funds of China (No. 12101333), and discovery grants of Wang and Nathoo from the Natural Sciences and Engineering Research Council of Canada RGPIN-2019-06131, RGPIN-06542-2014].

ORCID

Shufei Ge http://orcid.org/0000-0002-9395-3884
Shijia Wang http://orcid.org/0000-0003-0339-1716
Farouk S. Nathoo http://orcid.org/0000-0002-2569-3507
Liangliang Wang http://orcid.org/0000-0002-8509-7985

References

[1] James GM, Hastie TJ. Functional linear discriminant analysis for irregularly sampled curves. J R Stat Soc Ser B (Stat Methodol). 2001;63(3):533–550.
[2] Izenman AJ. Linear discriminant analysis. Modern multivariate statistical techniques. Springer; 2013. p. 237–280.
[3] Hall P, Poskitt DS, Presnell B. A functional data – analytic approach to signal discrimination. Technometrics. 2001;43(1):1–9.
[4] James GM, Sugar CA. Clustering for sparsely sampled functional data. J Am Stat Assoc. 2003;98(462):397–408.
[5] Müller H-G. Functional modelling and classification of longitudinal data. Scandinavian J Stat. 2005;32(2):223–240.
[6] Chamroukhi F, Samé A, Govaert G, et al. A hidden process regression model for functional data description. application to curve discrimination. Neurocomputing. 2010;73(7–9):1210–1221.
[7] Cheng G, Zhou L, Huang JZ, et al. Efficient semiparametric estimation in generalized partially linear additive models for longitudinal/clustered data. Bernoulli. 2014;20(1):141–163.
[8] Huang H, Li Y, Guan Y. Joint modeling and clustering paired generalized longitudinal trajectories with application to cocaine abuse treatment data. J Am Stat Assoc. 2014;109(508):1412–1424.

[9] Matheron G. Principles of geostatistics. Econ Geol. 1963;58(8):1246–1266.
[10] Duchon J. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. Constructive theory of functions of several variables. Springer; 1977. p. 85–100.
[11] Malfait N, Ramsay JO. The historical functional linear model. Canadian J Stat. 2003;31(2):115–128.
[12] Wood SN, Bravington MV, Hedley SL. Soap film smoothing. J R Stat Soc Ser B (Stat Methodol). 2008;70(5):931–955.
[13] Xun X, Cao J. Sparse estimation of historical functional linear models with a nested group bridge approach. preprint 2019. Available from: arXiv:1905.11676.
[14] Nguyen HD, McLachlan GJ, Wood IA. Mixtures of spatial spline regressions for clustering and classification. Comput Stat Data Anal. 2016;93:76–85.
[15] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Methodol). 1977;39:1–22.
[16] Srivastava S, DePalma G, Liu C. An asynchronous distributed expectation maximization algorithm for massive data: the DEM algorithm. J Comput Graph Stat. 2019;28(2):233–243.
[17] Diebolt J, Robert CP. Estimation of finite mixture distributions through Bayesian sampling. J R Stat Soc Ser B (Methodol). 1994;56:363–375.
[18] Jasra A, Holmes CC, Stephens DA. Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. Stat Sci. 2005;20:50–67.
[19] Chamroukhi F. Bayesian mixtures of spatial spline regressions. preprint 2015. Available from: arXiv:1508.00635.
[20] Chamroukhi F, Nguyen HD. Model-based clustering and classification of functional data. Wiley Interdisciplinary Rev Data Mining Knowledge Discovery. 2019;9(4):e1298.
[21] Doucet A, Godsill S, Andrieu C. On sequential Monte Carlo sampling methods for Bayesian filtering. Stat Comput. 2000;10(3):197–208.
[22] Doucet A, De Freitas N, Gordon N. An introduction to sequential Monte Carlo methods. Sequential Monte Carlo methods in practice. Springer; 2001. p. 3–14.
[23] Liu JS, Chen R. Sequential Monte Carlo methods for dynamic systems. J Am Stat Assoc. 1998;93(443):1032–1044.
[24] Carvalho CM, Lopes HF, Polson NG, et al. Particle learning for general mixtures. Bayesian Analysis. 2010;5(4):709–740.
[25] Fearnhead P, Meligkotsidou L. Filtering methods for mixture models. J Comput Graph Stat. 2007;16(3):586–607.
[26] MacEachern SN, Clyde M, Liu JS. Sequential importance sampling for nonparametric Bayes models: the next generation. Canadian J Stat. 1999;27(2):251–267.
[27] Carvalho CM, Johannes MS, Lopes HF, et al. Particle learning and smoothing. Stat Sci. 2010;25(1):88–106.
[28] Pitt MK, Shephard N. Filtering via simulation: auxiliary particle filters. J Am Stat Assoc. 1999;94(446):590–599.
[29] Fearnhead P. Markov chain Monte Carlo, sufficient statistics, and particle filters. J Comput Graph Stat. 2002;11(4):848–862.
[30] Gilks WR, Berzuini C. Following a moving target –Monte Carlo inference for dynamic Bayesian models. J R Stat Soc: Ser B (Stat Methodol). 2001;63(1):127–146.
[31] Sangalli LM, Ramsay JO, Ramsay TO. Spatial spline regression models. J R Stat Soc: Ser B (Stat Methodol). 2013;75(4):681–703.
[32] Hobert J, Casella G. The effect of improper priors on Gibbs sampling in hierarchical linear mixed models. J Am Stat Assoc. 1996;91(436):1461–1473.
[33] Doucet A, Briers M, Sénécal S. Efficient block sampling strategies for sequential Monte Carlo methods. J Comput Graph Stat. 2006;15(3):693–711.
[34] Lin M, Chen R, Liu JS, et al. Lookahead strategies for sequential Monte Carlo. Stat Sci. 2013;28(1):69–94.
[35] Stephens M. Dealing with label switching in mixture models. J R Stat Soc: Ser B (Stat Methodol). 2000;62(4):795–809.

[36] Geweke J. Interpretation and inference in mixture models: simple MCMC works. Comput Stat Data Anal. 2007;51(7):3529–3550.

[37] Puolamäki K, Kaski S. Bayesian solutions to the label switching problem. International Symposium on Intelligent Data Analysis; Springer; 2009. p. 381–392.

[38] Grün B, Leisch F. Dealing with label switching in mixture models under genuine multimodality. J Multivar Anal. 2009;100(5):851–861.

[39] Friel N, Pettitt AN. Marginal likelihood estimation via power posteriors. J R Stat Soc Ser B (Stat Methodol). 2008;70(3):589–607.

[40] Gelman A, Meng X-L. Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. Stat Sci. 1998;13:163–185.

[41] Gelman A, Hwang J, Vehtari A. Understanding predictive information criteria for Bayesian models. Stat Comput. 2014;24(6):997–1016.

[42] Watanabe S. Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory. J Mach Learn Res. 2010;11(12):3571–3594.

[43] Marin J-M, Robert C. Approximating the marginal likelihood in mixture models. preprint 2008. Available from: arXiv:0804.2414.

[44] Gelman A, John C, Hal S, et al. Bayesian data analysis. Chapman and Hall/CRC; 1995.

[45] Geweke J. Getting it right. J Am Stat Assoc. 2004;99:799–804.

[46] Wang L, Wang S, Bouchard-Côté A. An annealed sequential Monte Carlo method for Bayesian phylogenetics. Syst Biol. 2020;69(1):155–183.

[47] Novais L, Faria S. Comparison of the EM, CEM and SEM algorithms in the estimation of finite mixtures of linear mixed models: a simulation study. Comput Stat. 2021;27:1–27.

[48] Hubert L, Arabie P. Comparing partitions. J Classif. 1985;2(1):193–218.

[49] Friedman J, Hastie T, Tibshirani R. The elements of statistical learning. Springer; 2009. (Springer Series in Statistics; 1).

[50] Nathoo FS, Lesperance ML, Lawson AB, et al. Comparing variational Bayes with Markov Chain Monte Carlo for Bayesian computation in neuroimaging. Stat Methods Med Res. 2013;22(4):398–423.

[51] Nathoo FS, Babul A, Moiseev A, et al. A variational Bayes spatiotemporal model for electromagnetic brain mapping. Biometrics. 2014;70(1):132–143.

[52] Marin J-M, Mengersen K, Robert CP. Bayesian modelling and inference on mixtures of distributions. Handbook Statist. 2005;25:459–507.

[53] Chopin N. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. Ann Stat. 2004;32(6):2385–2411.

Appendices

Appendix 1. List of notations

Table A1. List of notations used in this paper.

Notation	Description
$^\top$	transpose symbol of a vector or matrix
T	total number of observations
N^*	total number of Gibbs sampling iterations.
N	total number of particles in online SMC algorithm
K	number of clusters
d	number of Nodal basis functions
k	index for cluster k, for $1 \leq k \leq K$
t	time index, takes value from $1 \leq t \leq T$
m_t	length of observation $\boldsymbol{y}_t$, for $1 \leq t \leq T$
$\boldsymbol{y}_t$	a $m_t \times 1$ vector, observation at time t, for $1 \leq t \leq T$
$\boldsymbol{\eta}$	$\boldsymbol{\eta} = (\eta_1, \eta_2)$, arbitrary coordinates
$\boldsymbol{\eta}_{t,1}, \ldots, \boldsymbol{\eta}_{t,m_t}$	coordinates for $\boldsymbol{y}_t$, for $1 \leq t \leq T$
$\boldsymbol{c}$	$\boldsymbol{c} = (c_1, c_2)$, centre parameter for a Nodal basis function
$\boldsymbol{c}_j$	$\boldsymbol{c}_j = (c_{j1}, c_{j2})$, centre parameter for j^{th} Nodal basis function, $1 \leq j \leq d$
$\boldsymbol{\delta}$	$\boldsymbol{\delta} = (\delta_1, \delta_2)$, shape parameter for a Nodal basis function
$s(\cdot)$	Nodal basis function
$\boldsymbol{S}_t$	a $m_t \times d$ matrix, spatial coordinates matrix of observation, $\boldsymbol{y}_t$, for $1 \leq t \leq T$
$\boldsymbol{\beta}_k$	a $d \times 1$ vector, fixed effects for cluster k, for $1 \leq k \leq K$
$\boldsymbol{b}_{tk}$	a $d \times 1$ vector, random effects of cluster k for $\boldsymbol{y}_t$, for $1 \leq k \leq K, 1 \leq t \leq T$
$\boldsymbol{e}_{tk}$	a $d \times 1$ vector, random error of cluster k for $\boldsymbol{y}_t$, for $1 \leq k \leq K, 1 \leq t \leq T$
$\boldsymbol{\sigma}_k^2, \boldsymbol{\xi}_k^2$	variance parameter for cluster k, $1 \leq k \leq K$
$\boldsymbol{\pi}$	$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, cluster allocation probability for observations
z_t	cluster label of $\boldsymbol{y}_t$ for $1 \leq t \leq T$
$\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{b}, \boldsymbol{b}_t$	$\boldsymbol{Y} = \{\boldsymbol{y}_t\}_{t=1}^T, \boldsymbol{Z} = \{z_t\}_{t=1}^T, \boldsymbol{b} = \{\boldsymbol{b}_t\}_{t=1}^T, \boldsymbol{b}_t = \{\boldsymbol{b}_{tk}\}_{k=1}^K$ for $1 \leq k \leq K$
$\boldsymbol{\theta}$	$\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\sigma}^2, \boldsymbol{\xi}^2\}$
$MVN, Dir, Mult, IG$	abbreviations for multivariate normal distribution, Dirichlet distribution, multinomial distribution and inverse-gamma distribution, respectively
$\phi(\cdot)$	density function of a multivariate normal distribution,
$f(\cdot), f_{\boldsymbol{\theta}}(\cdot), p(\cdot), p_{\boldsymbol{\theta}}(\cdot)$	density functions, usually $f(\cdot)(f_{\boldsymbol{\theta}}(\cdot))$ for a prior and $p(\cdot)(p_{\boldsymbol{\theta}}(\cdot))$ for a posterior, and $p(z_t = k)$ refers to probability of $z_t = k$ for $1 \leq k \leq K, 1 \leq t \leq T$
a_0, b_0, g_0, h_0	hyper parameters
$\{\alpha_k\}_{k=1}^K, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0$	hyper parameters
$\mathbf{1}(\cdot)$	indicator function
$\boldsymbol{b}_{1:t}, z_{1:t}, \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t}$	abbreviations of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_t, z_1, \ldots, z_t, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_t$, for $t = 1 \leq t \leq T$
$w_t^{(i)}, W_t^{(i)}$	unnormalized and normalized weight, respectively, for $1 \leq i \leq N, 1 \leq t \leq T$
$A_t^{(i)}$	ancestor index for particle i at time t for $1 \leq i \leq N, 1 \leq t \leq T$
$q_{t,\boldsymbol{\theta}}(\cdot)$	proposal distribution for $\boldsymbol{x}_t$ for $1 \leq t \leq T$
$\boldsymbol{s_t}$	sufficient statistics for the MSSR$_\mathrm{m}$ model given ($\boldsymbol{y}_{1:t}, z_{1:t}, \boldsymbol{b}_{1:t}$) for $1 \leq t \leq T$
$T(\cdot)$	function of $(\boldsymbol{s_{t-1}}, \boldsymbol{y}_t, z_t, \boldsymbol{b}_t)$, and takes $\boldsymbol{s_t}$ as return, for $2 \leq t \leq T$
$f_k(\cdot)$	function used to simulate surfaces from cluster k, $1 \leq k \leq K$

Appendix 2. Derivations for the Gibbs Sampling Algorithm.

In this section, we derived the full conditional distributions for $z_t, \boldsymbol{\pi}, \boldsymbol{\beta}_k, \boldsymbol{b}_{tk}, \boldsymbol{\sigma}_k^2$ and $\boldsymbol{\xi}_k^2$ and described the Gibbs sampler.

Given the hierarchical priors of the Bayesian mixture of spatial spline regression with mixed effects model described in Section 3.1, the full joint posterior distribution of $\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{\sigma}^2, \boldsymbol{\xi}^2, \boldsymbol{Z}$ can be

Algorithm 3 Markov chain Monte Carlo (MCMC) algorithm – Gibbs sampler.

```
Input: data $\boldsymbol{y}_{1:T}$, initial parameters $\{\boldsymbol{\theta}^{(0)\mathsf{T}}, \boldsymbol{b}^{(0)\mathsf{T}}\}^{\mathsf{T}}$, where $\boldsymbol{\theta}^{(0)\mathsf{T}} = (\boldsymbol{\pi}^{(0)\mathsf{T}}, \boldsymbol{\beta}^{(0)\mathsf{T}}, \boldsymbol{\sigma}^{2(0)\mathsf{T}}, \boldsymbol{\xi}^{2(0)\mathsf{T}})^{\mathsf{T}}$.
Output: $\{\boldsymbol{\theta}^{(i)\mathsf{T}}, \boldsymbol{b}^{(i)\mathsf{T}}\}^{\mathsf{T}}_{1\le i\le N^*}$, where $\boldsymbol{\theta}^{(i)\mathsf{T}} = (\boldsymbol{\pi}^{(i)\mathsf{T}}, \boldsymbol{\beta}^{(i)\mathsf{T}}, \boldsymbol{\sigma}^{2(i)\mathsf{T}}, \boldsymbol{\xi}^{2(i)\mathsf{T}})^{\mathsf{T}}$, $N^*$ is total number of iterations.
for $i = 1$ to $N^*$ do
  for $t = 1$ to $T$ do
    Sample $z_t^{(i)} \sim p(\cdot|\boldsymbol{y}_t, \boldsymbol{\pi}^{(i-1)}, \boldsymbol{\beta}^{(i-1)}, \boldsymbol{\sigma}^{2(i-1)}, \boldsymbol{\xi}^{2(i-1)}, \boldsymbol{b}^{(i-1)})$ according to Equation (A2) in Appendix.
  end for
  Sample $\boldsymbol{\pi}^{(\boldsymbol{i})} \sim p(\cdot|\boldsymbol{Y}, \boldsymbol{Z}^{(i)}, \boldsymbol{\beta}^{(i-1)}, \boldsymbol{\sigma}^{2(i-1)}, \boldsymbol{\xi}^{2(i-1)}, \boldsymbol{b}^{(i-1)})$ according to Equation (A3) in Appendix.
  for $k = 1$ to $K$ do
    Sample $\boldsymbol{\beta}_k^{(i)} \sim p(\cdot|\boldsymbol{Y}, \boldsymbol{Z}^{(i)}, \boldsymbol{\pi}^{(i)}, \sigma^{2(i-1)}_k, \xi^{2(i-1)}_k, \{\boldsymbol{b}_{tk}^{(i-1)}\}_{t=1}^T)$ according to Equation (A4) in Appendix.
    for $t = 1$ to $T$ do
      Sample $\boldsymbol{b}_{tk}^{(i)} \sim p(\cdot|\boldsymbol{Y}, \boldsymbol{Z}^{(i)}, \boldsymbol{\pi}^{(i)}, \boldsymbol{\beta}_k^{(i)}, \sigma^{2(i-1)}_k, \xi^{2(i-1)}_k)$ according to Equation (A5) in Appendix.
    end for
    Sample $\sigma_k^{2(i)} \sim p(\cdot|\boldsymbol{Y}, \boldsymbol{Z}^{(i)}, \boldsymbol{\pi}^{(i)}, \boldsymbol{\beta}_k^{(i)}, \xi^{2(i-1)}_k, \{\boldsymbol{b}_{tk}^{(i)}\}_{t=1}^T)$ according to Equation (A6) in Appendix.
    Sample $\xi_k^{2(i)} \sim p(\cdot|\boldsymbol{Y}, \boldsymbol{Z}^{(i)}, \boldsymbol{\pi}^{(i)}, \boldsymbol{\beta}_k^{(i)}, \sigma^{2(i)}_k, \{\boldsymbol{b}_{tk}^{(i)}\}_{t=1}^T)$ according to Equation (A7) in Appendix.
  end for
end for
```

expressed up to a marginal likelihood as

$$
\begin{aligned}
p(\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{\sigma}^2, \boldsymbol{\xi}^2, \boldsymbol{Z}|\boldsymbol{Y}) &\propto f(\boldsymbol{\pi}) f(\boldsymbol{\beta}^2) f(\boldsymbol{\xi}^2) f(\boldsymbol{\sigma}^2) f(\boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{\xi}^2, \boldsymbol{\sigma}^2) \\
&\propto f(\boldsymbol{\pi}) \times \prod_{k=1}^{K} f(\boldsymbol{\beta}_k) \times \prod_{k=1}^{K} f(\xi_k^2) \times \prod_{t=1}^{T}\prod_{k=1}^{K} f(\boldsymbol{b}_{tk}|\xi_k^2)) \times \prod_{k=1}^{K} f(\sigma_k^2) \\
&\quad \times \prod_{t=1}^{T}\prod_{k=1}^{K} \left\{\phi(\boldsymbol{y}_t|\boldsymbol{S}_t\boldsymbol{\beta}_k + \boldsymbol{S}_t\boldsymbol{b}_{tk}, \sigma_k^2\boldsymbol{I}_{m_t}) p(z_t = k)\right\}^{\mathbf{1}_k(z_t)} \\
&\propto f(\boldsymbol{\pi}) \times \prod_{k=1}^{K} f(\boldsymbol{\beta}_k) f(\xi_k^2) f(\sigma_k^2) \times \prod_{t=1}^{T}\prod_{k=1}^{K} f(\boldsymbol{b}_{tk}|\xi_k^2)) \\
&\quad \times \prod_{t=1}^{T}\prod_{k=1}^{K} \left\{\phi(\boldsymbol{y}_t|\boldsymbol{S}_t\boldsymbol{\beta}_k + \boldsymbol{S}_t\boldsymbol{b}_{tk}, \sigma_k^2\boldsymbol{I}_{m_t}) p(z_t = k)\right\}^{\mathbf{1}_k(z_t)} \\
&\propto f(\boldsymbol{\pi}) \times \prod_{k=1}^{K} f(\boldsymbol{\beta}_k) f(\xi_k^2) f(\sigma_k^2) (\xi_k^2)^{-\frac{dT}{2}}
\end{aligned}
$$

$$\begin{aligned}
&\times \prod_{k=1}^{K} \exp\{-\frac{1}{2\xi_k^2} \sum_{t=1}^{T} \boldsymbol{b}_{tk}^{\mathsf{T}} \boldsymbol{b}_{tk} - \log(\sigma_k^2)\frac{d}{2} \sum_{t=1}^{T} \mathbf{1}_k(z_t) + \log(\pi_k) \sum_{t=1}^{T} \mathbf{1}_k(z_t)\} \\
&\times \prod_{k=1}^{K} \exp\{-\frac{1}{2\sigma_k^2}[\sum_{t=1}^{T} \mathbf{1}_k(z_t)(\boldsymbol{y}_t - \boldsymbol{S}_t\boldsymbol{b}_{tk})^{\mathsf{T}}(\boldsymbol{y}_t - \boldsymbol{S}_t\boldsymbol{b}_{tk}) \\
&- 2\sum_{t=1}^{T} \mathbf{1}_k(z_t)\boldsymbol{\beta}_k \boldsymbol{S}_t^{\mathsf{T}}(\boldsymbol{y}_t - \boldsymbol{S}_t\boldsymbol{b}_{tk})) \\
&+ \sum_{t=1}^{T} \mathbf{1}_k(z_t)(\boldsymbol{\beta}_k^{\mathsf{T}} \boldsymbol{S}_t^{\mathsf{T}} \boldsymbol{S}_t \boldsymbol{\beta}_k)]\}. \qquad \text{(A1)}
\end{aligned}$$

Full conditional distribution of z_t

$$p(z_t|\boldsymbol{Y}, \boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{\sigma}^2, \boldsymbol{\xi}^2) \propto \prod_{k=1}^{K} \{\phi(\boldsymbol{y}_t; \boldsymbol{S}_t\boldsymbol{\beta}_k + S_i\boldsymbol{b}_{tk}, \sigma_k^2 \boldsymbol{I}_{m_t}) p(z_t = k)\}^{\mathbf{1}_k(z_t)}.$$

Therefore,

$$z_t = k|\boldsymbol{Y}, \boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{\sigma}^2, \boldsymbol{\xi}^2 \sim Mult(1; \tau_{t1}, \ldots, \tau_{tK}), \qquad \text{(A2)}$$

where

$$\tau_{tk} = \frac{\phi(\boldsymbol{y}_t; \boldsymbol{S}_t\boldsymbol{\beta}_k + \boldsymbol{S}_t\boldsymbol{b}_{tk}, \sigma_k^2 \boldsymbol{I}_{m_t})\pi_k}{\sum_{j=1}^{K} \phi(\boldsymbol{y}_t; \boldsymbol{S}_t\boldsymbol{\beta}_j + \boldsymbol{S}_t\boldsymbol{b}_{tj}, \sigma_j^2 \boldsymbol{I}_{m_t})\pi_j},$$

for $1 \le k \le K,\ 1 \le t \le T$.

Full conditional distribution of π

$$\begin{aligned}
p(\boldsymbol{\pi}|\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{\sigma}^2, \boldsymbol{\xi}^2) &\propto p(\boldsymbol{\pi}) \prod_{t=1}^{T} \prod_{k=1}^{K} p(z_t = k)^{\mathbf{1}_k(z_t)} \\
&\propto \prod_{k=1}^{K} \pi_k^{(\alpha_k + n_{T,k}) - 1}.
\end{aligned}$$

Therefore,

$$\boldsymbol{\pi}|\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{\sigma}^2, \boldsymbol{\xi}^2 \sim Dir(\alpha_1 + n_{T,1}, \ldots, \alpha_K + n_{T,K}), \qquad \text{(A3)}$$

where $n_{T,k} = \sum_{t=1}^{T} \mathbf{1}_k(z_t)$, for$1 \le k \le K$.

Full conditional distribution of β_k

$$\begin{aligned}
p(\boldsymbol{\beta}_k|\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{b}, \boldsymbol{\sigma}^2, \boldsymbol{\xi}^2) &\propto f(\boldsymbol{\beta}_k|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \prod_{t=1}^{T} \left\{\phi(\boldsymbol{y}_t|\boldsymbol{S}_t\boldsymbol{\beta}_k + S_t\boldsymbol{b}_{tk}, \sigma_k^2 \boldsymbol{I}_{m_t})\right\}^{\mathbf{1}_k(z_t)} \\
&\propto \exp\left\{-\frac{1}{2}\boldsymbol{\beta}_k^{\mathsf{T}} \left(\boldsymbol{\Sigma}_0^{-1} + \sum_{t=1}^{T} \mathbf{1}_k(z_t)\frac{\boldsymbol{S}_t^{\mathsf{T}}\boldsymbol{S}_t}{\sigma_k^2}\right)\boldsymbol{\beta}_k\right\} \\
&\times \exp\left\{\boldsymbol{\beta}_k^{\mathsf{T}} \left(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 + \sum_{t=1}^{T} \mathbf{1}_k(z_t)\frac{\boldsymbol{S}_t^{\mathsf{T}}(\boldsymbol{y}_t - \boldsymbol{S}_t\boldsymbol{b}_{tk})}{\sigma_k^2}\right)\right\}.
\end{aligned}$$

Therefore,

$$\boldsymbol{\beta}_k | \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{b}, \boldsymbol{\sigma}^2, \boldsymbol{\xi}^2 \sim MVN(\boldsymbol{\mu}_{\boldsymbol{\beta}_k}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_k}), \tag{A4}$$

where $1 \le k \le K$, $\boldsymbol{\mu}_{\boldsymbol{\beta}_k} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_k}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 + \sum_{t=1}^T \mathbf{1}_k(z_t)\boldsymbol{S}_t^\top(\boldsymbol{y}_t - \boldsymbol{S}_t\boldsymbol{b}_{tk})/\sigma_k^2)$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_k}^{-1} = \boldsymbol{\Sigma}_0^{-1} + \sum_{t=1}^T \mathbf{1}_k(z_t)\boldsymbol{S}_t^\top \boldsymbol{S}_t/\sigma_k^2$.

Full conditional distribution of $\boldsymbol{b}_{tk}$

$$\begin{aligned}
&p(\boldsymbol{b}_{tk} | \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\sigma}^2, \boldsymbol{\xi}^2) \\
&\quad \propto f(\boldsymbol{b}_{tk} | 0_d, \xi_k^2 \boldsymbol{I}_d)\phi(\boldsymbol{y}_t; \boldsymbol{S}_t\boldsymbol{\beta}_k + \boldsymbol{S}_t\boldsymbol{b}_{tk}, \sigma_k^2 \boldsymbol{I}_{m_t})^{\mathbf{1}_k(z_t)} \\
&\quad \propto \exp\left\{-\frac{1}{2}\left[\boldsymbol{b}_{tk}^\top \left(\frac{1}{\xi_k^2}\boldsymbol{I}_d + \mathbf{1}_k(z_t)\frac{\boldsymbol{S}_t^\top \boldsymbol{S}_t}{\sigma_k^2}\right)\boldsymbol{b}_{tk} - 2\boldsymbol{b}_{tk}^\top \mathbf{1}_k(z_t)\frac{\boldsymbol{S}_t^\top(\boldsymbol{y}_t - \boldsymbol{S}_t\boldsymbol{\beta}_k)}{\sigma_k^2}\right]\right\}.
\end{aligned}$$

Therefore,

$$\boldsymbol{b}_{tk} | \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\sigma}^2, \boldsymbol{\xi}^2 \sim MVN(\boldsymbol{\mu}_{\boldsymbol{b}_{tk}}, \boldsymbol{\Sigma}_{\boldsymbol{b}_{tk}}), \tag{A5}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{b}_{tk}}^{-1} = \frac{1}{\xi_k^2}\boldsymbol{I}_d + \mathbf{1}_k(z_t)\frac{\boldsymbol{S}_t^\top \boldsymbol{S}_t}{\sigma_k^2}$, and $\boldsymbol{\mu}_{\boldsymbol{b}_{tk}} = \mathbf{1}_k(z_t)\boldsymbol{\Sigma}_{\boldsymbol{b}_{tk}}\frac{\boldsymbol{S}_t^\top(\boldsymbol{y}_t - \boldsymbol{S}_t\boldsymbol{\beta}_k)}{\sigma_k^2}$, for $1 \le t \le T, 1 \le k \le K$.

Full conditional distribution of σ_k^2

$$\begin{aligned}
p(\sigma_k^2 | \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{\xi}^2) &\propto f(\sigma_k^2) f(\boldsymbol{Y} | \boldsymbol{Z}, \boldsymbol{\beta}_k, \boldsymbol{b}_{tk}, \sigma_k^2, \xi_k^2, \pi_k) \\
&\propto (\sigma_k^2)^{-\left\{g_0 + \frac{m_t}{2}\sum_{t=1}^T \mathbf{1}_k(z_t)\right\} - 1} \\
&\quad \times \exp\left\{-\frac{h_0 + \frac{\sum_{t=1}^T \mathbf{1}_k(z_t)(\boldsymbol{y}_t - \boldsymbol{S}_t\boldsymbol{\beta}_k - \boldsymbol{S}_t\boldsymbol{b}_{tk})^\top(\boldsymbol{y}_t - \boldsymbol{S}_t\boldsymbol{\beta}_k - \boldsymbol{S}_t\boldsymbol{b}_{tk})}{2}}{\sigma_k^2}\right\}.
\end{aligned}$$

Therefore,

$$\boldsymbol{\sigma}_k^2 | \boldsymbol{Y}, \quad \boldsymbol{Z}, \quad \boldsymbol{\pi}, \quad \boldsymbol{b}, \quad \boldsymbol{\beta}, \quad \boldsymbol{\xi}^2 \sim IG(g_0^*, h_0^*), \tag{A6}$$

where $g_0^* = g_0 + \frac{n_{T,k}}{2}m_t, h_0^* = h_0 + \frac{\sum_{t=1}^T \mathbf{1}_k(z_t)(\boldsymbol{y}_t - \boldsymbol{S}_t\boldsymbol{\beta}_k - \boldsymbol{S}_t\boldsymbol{b}_{tk})^\top(\boldsymbol{y}_t - \boldsymbol{S}_t\boldsymbol{\beta}_k - \boldsymbol{S}_t\boldsymbol{b}_{tk})}{2}, n_{T,k} = \sum_{t=1}^T \mathbf{1}_k(z_t)$, for $1 \le k \le K$.

Full conditional distribution of ξ_k^2

$$\begin{aligned}
p(\xi_k^2 | \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{\sigma}^2) &\propto f(\xi_k^2)\prod_{t=1}^T f(\boldsymbol{b}_{tk} | \xi_k^2) \\
&\propto (\xi_k^2)^{-(a_0 + \frac{nd}{2}) - 1}\exp\left\{-\frac{b_0 + \frac{1}{2}\sum_{t=1}^T \boldsymbol{b}_{tk}^\top \boldsymbol{b}_{tk}}{\xi_k^2}\right\}.
\end{aligned}$$

Therefore,

$$\xi_k^2 | \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{b}, \boldsymbol{\beta}, \boldsymbol{\sigma}^2 \sim IG(a_0^*, b_0^*), \tag{A7}$$

where $a_0^* = a_0 + \frac{Td}{2}$, $b_0^* = b_0 + \frac{1}{2}\sum_{t=1}^T \boldsymbol{b}_{tk}^\top \boldsymbol{b}_{tk}$, for $1 \le k \le K$.

Appendix 3. Proof of Proposition 3.1

As indicated by Equation (A1) in A, the joint posterior distribution of $\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\sigma}^2$, conditional on $z_{1:t}, \boldsymbol{b}_{1:t}, \boldsymbol{y}_{1:t}$ can be expressed up to a marginal likelihood as

$$
\begin{aligned}
&p(\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\sigma}^2, \boldsymbol{\xi}^2 | z_{1:t}, \boldsymbol{b}_{1:t}, \boldsymbol{y}_{1:t}) \propto f(\boldsymbol{\pi}) f(\boldsymbol{\beta}^2) f(\boldsymbol{\xi}^2) f(\boldsymbol{\sigma}^2) f(\boldsymbol{y}_{1:t}, z_{1:t} | \boldsymbol{\beta}, \boldsymbol{b}_{1:t}, \boldsymbol{\xi}^2, \boldsymbol{\sigma}^2) \\
&\propto f(\boldsymbol{\pi}) \times \prod_{k=1}^{K} f(\boldsymbol{\beta}_k) f(\xi_k^2) f(\sigma_k^2) (\xi_k^2)^{-\frac{\mathrm{d}t}{2}} \\
&\times \prod_{k=1}^{K} \exp\{-\frac{1}{2\xi_k^2} \sum_{t=1}^{t} \boldsymbol{b}_{t'k}^{\mathsf{T}} \boldsymbol{b}_{t'k} - \log(\sigma_k^2) \frac{d}{2} \sum_{t=1}^{t} \mathbf{1}_k(z_{t'}) + \log(\pi_k) \sum_{t'=1}^{t} \mathbf{1}_k(z_{t'})\} \\
&\times \prod_{k=1}^{K} \exp\{-\frac{1}{2\sigma_k^2} \sum_{t'=1}^{t} \mathbf{1}_k(z_{t'}) [(\boldsymbol{y}_{t'} - \boldsymbol{S}_{t'} \boldsymbol{b}_{t'k})^{\mathsf{T}} (\boldsymbol{y}_{t'} - \boldsymbol{S}_{t'} \boldsymbol{b}_{t'k}) \\
&- 2\boldsymbol{\beta}_k \boldsymbol{S}_{t'}^{\mathsf{T}} (\boldsymbol{y}_{t'} - \boldsymbol{S}_{t'} \boldsymbol{b}_{t'k}) + \boldsymbol{\beta}_k^{\mathsf{T}} \boldsymbol{S}_{t'}^{\mathsf{T}} \boldsymbol{S}_{t'} \boldsymbol{\beta}_k]\} \\
&\propto f(\boldsymbol{\pi}) \times \prod_{k=1}^{K} f(\boldsymbol{\beta}_k) f(\xi_k^2) f(\sigma_k^2) (\xi_k^2)^{-\frac{\mathrm{d}t}{2}} \\
&\times \prod_{k=1}^{K} \exp\{-\frac{1}{2\xi_k^2} \sum_{t'=1}^{t} \boldsymbol{b}_{t'k}^{\mathsf{T}} \boldsymbol{b}_{t'k} - \log(\sigma_k^2) \frac{\mathrm{d}}{2} \sum_{t'=1}^{t} \mathbf{1}_k(z_{t'}) + \log(\pi_k) \sum_{t'=1}^{t} \mathbf{1}_k(z_{t'})\} \\
&\times \prod_{k=1}^{K} \exp\{-\frac{1}{2\sigma_k^2} [\sum_{t'=1}^{t} \mathbf{1}_k(z_{t'}) (\boldsymbol{y}_{t'} - \boldsymbol{S}_{t'} \boldsymbol{b}_{t'k})^{\mathsf{T}} (\boldsymbol{y}_{t'} - \boldsymbol{S}_{t'} \boldsymbol{b}_{t'k}) \\
&- 2\boldsymbol{\beta}_k \sum_{t'=1}^{t} \mathbf{1}_k(z_{t'}) \boldsymbol{S}_{t'}^{\mathsf{T}} (\boldsymbol{y}_{t'} - \boldsymbol{S}_{t'} \boldsymbol{b}_{t'k}) + \boldsymbol{\beta}_k^{\mathsf{T}} (\sum_{t'=1}^{t} \mathbf{1}_k(z_{t'}) \boldsymbol{S}_{t'}^{\mathsf{T}} \boldsymbol{S}_{t'}) \boldsymbol{\beta}_k]\}.
\end{aligned}
$$

Denote $\boldsymbol{y}_{t'}^* = \boldsymbol{y}_{t'} - \boldsymbol{S}_{t'} \boldsymbol{b}_{t'k}$, the sufficient statistics $\boldsymbol{s_t}$ for the MSSR$_\mathrm{m}$ model given $(\boldsymbol{y}_{1:t}, z_{1:t}, \boldsymbol{b}_{1:t})$ for $1 \le t \le T$ can be written as

$$
\boldsymbol{s_t} = \left\{ \sum_{t'=1}^{t} \boldsymbol{b}_{t'k}^{\mathsf{T}} \boldsymbol{b}_{t'k}, \sum_{t'=1}^{t} \mathbf{1}_k(z_{t'}), \sum_{t'=1}^{t} \mathbf{1}_k(z_{t'}) \boldsymbol{S}_{t'}^{\mathsf{T}} \boldsymbol{y}_{t'}^*, \sum_{t'=1}^{t} \mathbf{1}_k(z_{t'}) \boldsymbol{y}_{t'}^{*\mathsf{T}} \boldsymbol{y}_{t'}^*, \sum_{t'=1}^{t} \mathbf{1}_k(z_{t'}) \boldsymbol{S}_{t'}^{\mathsf{T}} \boldsymbol{S}_{t'} \right\}_{k=1}^{K}.
$$

Appendix 4. Proof of Proposition 4.1

Our online SMC algorithm is run for time steps $t = 1, \ldots, T$, samples the random variables $\boldsymbol{\theta}_t = \{\boldsymbol{\theta}_t^{(i)}\}_{i=1}^N$, $\boldsymbol{x}_t = \{\boldsymbol{x}_t^{(i)}\}_{i=1}^N$, $\boldsymbol{A}_t = \{A_t^{(i)}\}_{i=1}^N$. The distributions of these random variables are:

$$A_t^{(i)} \sim W_{t-1}^{(i)}, \tag{A8}$$

$$\boldsymbol{x}_t^{(i)} \sim p(\boldsymbol{x}_t | \boldsymbol{\theta}_{t-1}^{A_t^{(i)}}, \boldsymbol{y}_t), \tag{A9}$$

$$\boldsymbol{\theta}_t^{(i)} \sim p(\boldsymbol{\theta}_t | \boldsymbol{x}_{1:t}^{(i)}, \boldsymbol{y}_{1:t}). \tag{A10}$$

Note that Equation A10 is true only if the Gibbs chain for $\boldsymbol{\theta}_t^{(i)}$ reaches stationary.

The distribution of all random variables is

$$\begin{aligned}\psi_{N,T}(\boldsymbol{x}_{1:T-1},\boldsymbol{\theta}_{0:T-1},\boldsymbol{A}_{1:T-1}) = \left\{\prod_{i=1}^{N} p(\boldsymbol{\theta}_0^{(i)})\right\}\prod_{t=1}^{T-1}\left\{\prod_{i=1}^{N} W_{t-1}^{A_t^{(i)}} p(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1}^{A_t^{(i)}},\boldsymbol{y}_t)\right.\\ \left.\times p(\boldsymbol{\theta}_t|\boldsymbol{x}_{1:t}^{(i)},\boldsymbol{y}_{1:t})\right\}.\end{aligned} \tag{A11}$$

To prove

$$E_{\psi_{N,T}(\boldsymbol{x}_{1:T-1},\boldsymbol{\theta}_{0:T-1},\boldsymbol{A}_{1:T-1})}\left(\prod_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{y}_t|\boldsymbol{\theta}_{t-1}^{(i)})\right) = p(\boldsymbol{y}_{1:T}), \tag{A12}$$

we solve the integral

$$\begin{aligned}
&E_{\psi_{N,T}(\boldsymbol{x}_{1:T-1},\boldsymbol{\theta}_{0:T-1},\boldsymbol{A}_{1:T-1})}\left(\prod_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{y}_t|\boldsymbol{\theta}_{t-1}^{(i)})\right)\\
&= \int \prod_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{y}_t|\boldsymbol{\theta}_{t-1}^{(i)})\left\{\prod_{i=1}^{N} p(\boldsymbol{\theta}_0^{(i)})\right\}\prod_{t=1}^{T-1}\left\{\prod_{i=1}^{N} W_{t-1}^{A_t^{(i)}} p(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1}^{A_t^{(i)}},\boldsymbol{y}_t)\right.\\
&\quad \left.\times p(\boldsymbol{\theta}_t|\boldsymbol{x}_{1:t}^{(i)},\boldsymbol{y}_{1:t})\right\}\,\mathrm{d}\boldsymbol{x}_{1:T-1}\,\mathrm{d}\boldsymbol{\theta}_{0:T-1}\,\mathrm{d}\boldsymbol{A}_{1:T-1}\\
&= \int\left\{\int \frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{y}_T|\boldsymbol{\theta}_{T-1}^{(i)})\left\{\prod_{i=1}^{N} W_{T-2}^{A_{T-1}^{(i)}} p(\boldsymbol{x}_{T-1}|\boldsymbol{\theta}_{T-2}^{A_{T-1}^{(i)}},\boldsymbol{y}_{T-1})\right.\right.\\
&\quad \left.\left.\times p(\boldsymbol{\theta}_{T-1}|\boldsymbol{x}_{1:T-1}^{(i)},\boldsymbol{y}_{1:T-1})\right\}\,\mathrm{d}\boldsymbol{x}_{T-1}\,\mathrm{d}\boldsymbol{\theta}_{T-1}\,\mathrm{d}\boldsymbol{A}_{T-1}\right\}\\
&\quad \times \prod_{t=1}^{T-1}\frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{y}_t|\boldsymbol{\theta}_{t-1}^{(i)})\left\{\prod_{i=1}^{N} p(\boldsymbol{\theta}_0^{(i)})\right\}\\
&\quad \times \prod_{t=1}^{T-2}\left\{\prod_{i=1}^{N} W_{t-1}^{A_t^{(i)}} p(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1}^{A_t^{(i)}},\boldsymbol{y}_t) p(\boldsymbol{\theta}_t|\boldsymbol{x}_{1:t}^{(i)},\boldsymbol{y}_{1:t})\right\}\,\mathrm{d}\boldsymbol{x}_{1:T-2}\,\mathrm{d}\boldsymbol{\theta}_{0:T-2}\,\mathrm{d}\boldsymbol{A}_{1:T-2}.
\end{aligned}$$

We first consider the integral over $(\boldsymbol{x}_{T-1},\boldsymbol{\theta}_{T-1},\boldsymbol{A}_{T-1})$,

$$\begin{aligned}
&\int \frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{y}_T|\boldsymbol{\theta}_{T-1}^{(i)})\left\{\prod_{i=1}^{N} W_{T-2}^{A_{T-1}^{(i)}} p(\boldsymbol{x}_{T-1}|\boldsymbol{\theta}_{T-2}^{A_{T-1}^{(i)}},\boldsymbol{y}_{T-1})\right.\\
&\qquad \left.\times p(\boldsymbol{\theta}_{T-1}|\boldsymbol{x}_{1:T-1}^{(i)},\boldsymbol{y}_{1:T-1})\right\}\,\mathrm{d}\boldsymbol{x}_{T-1}\,\mathrm{d}\boldsymbol{\theta}_{T-1}\,\mathrm{d}\boldsymbol{A}_{T-1}\\
&\quad = \sum_{i=1}^{N} W_{T-2}^{(i)} p(\boldsymbol{y}_T|\boldsymbol{\theta}_{T-2}^{(i)},\boldsymbol{x}_{1:T-2}^{(i)},\boldsymbol{y}_{1:T-1}).
\end{aligned}$$

Hence,

$$
\begin{aligned}
&E_{\psi_{N,T}(\boldsymbol{x}_{1:T-1},\boldsymbol{\theta}_{0:T-1},\boldsymbol{A}_{1:T-1})}\left(\prod_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N}p(\boldsymbol{y}_t|\boldsymbol{\theta}_{t-1}^{(i)})\right)\\
&\quad=\int\Bigg\{\int\sum_{i=1}^{N}W_{T-2}^{(i)}p(\boldsymbol{y}_T|\boldsymbol{\theta}_{T-2}^{(i)},\boldsymbol{x}_{1:T-2}^{(i)},\boldsymbol{y}_{1:T-1})\frac{1}{N}\sum_{i=1}^{N}p(\boldsymbol{y}_{T-1}|\boldsymbol{\theta}_{T-2}^{(i)})\\
&\qquad\times\Bigg\{\prod_{i=1}^{N}W_{T-3}^{A_{T-2}^{(i)}}p(\boldsymbol{x}_{T-2}|\boldsymbol{\theta}_{T-3}^{A_{T-2}^{(i)}},\boldsymbol{y}_{T-2})\\
&\qquad\times p(\boldsymbol{\theta}_{T-2}|\boldsymbol{x}_{1:T-2}^{(i)},\boldsymbol{y}_{1:T-2})\Bigg\}\ \mathrm{d}\boldsymbol{x}_{T-2}\,\mathrm{d}\boldsymbol{\theta}_{T-2}\,\mathrm{d}\boldsymbol{A}_{T-2}\Bigg\}\\
&\qquad\times\prod_{t=1}^{T-2}\frac{1}{N}\sum_{i=1}^{N}p(\boldsymbol{y}_t|\boldsymbol{\theta}_{t-1}^{(i)})\left\{\prod_{i=1}^{N}p(\boldsymbol{\theta}_0^{(i)})\right\}\prod_{t=1}^{T-3}\Bigg\{\prod_{i=1}^{N}W_{t-1}^{A_t^{(i)}}p(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1}^{A_t^{(i)}},\boldsymbol{y}_t)\\
&\qquad\times p(\boldsymbol{\theta}_t|\boldsymbol{x}_{1:t}^{(i)},\boldsymbol{y}_{1:t})\Bigg\}\ \mathrm{d}\boldsymbol{x}_{1:T-2}\,\mathrm{d}\boldsymbol{\theta}_{0:T-2}\,\mathrm{d}\boldsymbol{A}_{1:T-2}.
\end{aligned}
$$

We then consider the integral over $(\boldsymbol{x}_{T-2},\boldsymbol{\theta}_{T-2},\boldsymbol{A}_{T-2})$,

$$
\begin{aligned}
&\int\sum_{i=1}^{N}W_{T-2}^{(i)}p(\boldsymbol{y}_T|\boldsymbol{\theta}_{T-2}^{(i)},\boldsymbol{x}_{1:T-2}^{(i)},\boldsymbol{y}_{1:T-1})\frac{1}{N}\sum_{i=1}^{N}p(\boldsymbol{y}_{T-1}|\boldsymbol{\theta}_{T-2}^{(i)})\\
&\qquad\times\Bigg\{\prod_{i=1}^{N}W_{T-3}^{A_{T-2}^{(i)}}p(\boldsymbol{x}_{T-2}|\boldsymbol{\theta}_{T-3}^{A_{T-2}^{(i)}},\boldsymbol{y}_{T-2})\\
&\qquad\times p(\boldsymbol{\theta}_{T-2}|\boldsymbol{x}_{1:T-2}^{(i)},\boldsymbol{y}_{1:T-2})\Bigg\}\ \mathrm{d}\boldsymbol{x}_{T-2}\,\mathrm{d}\boldsymbol{\theta}_{T-2}\,\mathrm{d}\boldsymbol{A}_{T-2}\\
&\quad=\int\frac{1}{N}\sum_{i=1}^{N}p(\boldsymbol{y}_{T-1}|\boldsymbol{\theta}_{T-2}^{(i)})p(\boldsymbol{y}_T|\boldsymbol{\theta}_{T-2}^{(i)},\boldsymbol{x}_{1:T-2}^{(i)},\boldsymbol{y}_{1:T-1})\\
&\qquad\times\Bigg\{\prod_{i=1}^{N}W_{T-3}^{A_{T-2}^{(i)}}p(\boldsymbol{x}_{T-2}|\boldsymbol{\theta}_{T-3}^{A_{T-2}^{(i)}},\boldsymbol{y}_{T-2})\\
&\qquad\times p(\boldsymbol{\theta}_{T-2}|\boldsymbol{x}_{1:T-2}^{(i)},\boldsymbol{y}_{1:T-2})\Bigg\}\ \mathrm{d}\boldsymbol{x}_{T-2}\,\mathrm{d}\boldsymbol{\theta}_{T-2}\,\mathrm{d}\boldsymbol{A}_{T-2}\\
&\quad=\int\frac{1}{N}\sum_{i=1}^{N}p(\boldsymbol{y}_{T-1:T}|\boldsymbol{\theta}_{T-2}^{(i)},\boldsymbol{x}_{1:T-2}^{(i)},\boldsymbol{y}_{1:T-2})\\
&\qquad\times\Bigg\{\prod_{i=1}^{N}W_{T-3}^{A_{T-2}^{(i)}}p(\boldsymbol{x}_{T-2}|\boldsymbol{\theta}_{T-3}^{A_{T-2}^{(i)}},\boldsymbol{y}_{T-2})\\
&\qquad\times p(\boldsymbol{\theta}_{T-2}|\boldsymbol{x}_{1:T-2}^{(i)},\boldsymbol{y}_{1:T-2})\Bigg\}\ \mathrm{d}\boldsymbol{x}_{T-2}\,\mathrm{d}\boldsymbol{\theta}_{T-2}\,\mathrm{d}\boldsymbol{A}_{T-2}\\
&\quad=\sum_{i=1}^{N}W_{T-3}^{(i)}p(\boldsymbol{y}_{T-1:T}|\boldsymbol{\theta}_{T-3}^{(i)},\boldsymbol{x}_{1:T-3}^{(i)},\boldsymbol{y}_{1:T-2}).
\end{aligned}
$$

Similarly, we can integrate over $(\boldsymbol{x}_t, \boldsymbol{\theta}_t, \boldsymbol{A}_t)$ $(t = T-3, T-4, \dots, 1)$ and get

$$
\begin{aligned}
&E_{\psi_{N,T}(\boldsymbol{x}_{1:T-1},\boldsymbol{\theta}_{0:T-1},\boldsymbol{A}_{1:T-1})}\left(\prod_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N}p(\boldsymbol{y}_t|\boldsymbol{\theta}_{t-1}^{(i)})\right)\\
&= \int \sum_{i=1}^{N} W_0^{(i)} p(\boldsymbol{y}_{2:T}|\boldsymbol{\theta}_0^{(i)}, \boldsymbol{y}_1) \times \frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{y}_1|\boldsymbol{\theta}_0^{(i)}) \times \prod_{i=1}^{N} p(\boldsymbol{\theta}_0^{(i)})\, \mathrm{d}\boldsymbol{\theta}_0\\
&= \int \frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{y}_1|\boldsymbol{\theta}_0^{(i)}) p(\boldsymbol{y}_{2:T}|\boldsymbol{\theta}_0^{(i)}, \boldsymbol{y}_1) \times \prod_{i=1}^{N} p(\boldsymbol{\theta}_0^{(i)})\, \mathrm{d}\boldsymbol{\theta}_0\\
&= \int \frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta}_0^{(i)}) \times \prod_{i=1}^{N} p(\boldsymbol{\theta}_0^{(i)})\, \mathrm{d}\boldsymbol{\theta}_0\\
&= p(\boldsymbol{y}_{1:T}).
\end{aligned}
$$

This proves the proposition.

Appendix 5. Results of simulation study

As shown in Figures A1–A2, the estimated mean surfaces of proposed online SMC algorithm (common features) for each cluster is comparable as that of the MCMC algorithm.

As indicated in Figure A3, online SMC algorithm achieves similar performance as the MCMC algorithm in terms of $\xi_1^2, \xi_2^2, \xi_3^2$.

As indicated in Figure A4, online SMC algorithm achieves similar performance as the MCMC algorithm in terms of $\xi_1^2, \xi_2^2, \xi_3^2, \xi_4^2$. And there exists a small bias between the true value and the posterior mean provided by online SMC method for ξ_5^2, ξ_6^2 and the bias gets smaller as t increases.

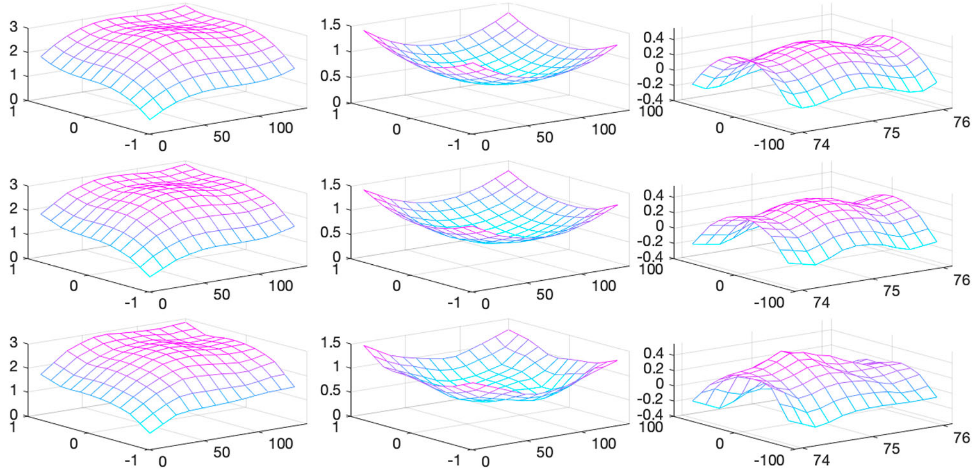

Figure A1. True surfaces versus estimated common features based on MSSR_m model: the top row displays the true surfaces of simulated by functions $f_1, \dots, f_3$ uniformly over the rectangular domain $[-1, 1] \times [-1, 1]$, the middle row and bottom row are the corresponding MCMC fitted mean surfaces and online SMC fitted mean surfaces at $T = 5000$ with $d = 6 \times 6$, $n.\min = 20$.

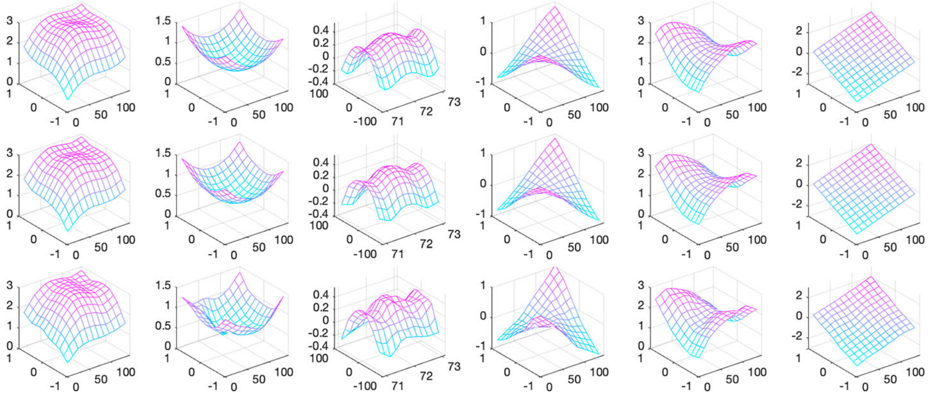

Figure A2. True surfaces versus estimated common features based on $\mathrm{MSSR_m}$ model: the top row displays the true surfaces of simulated by functions $f_1, \ldots, f_6$ uniformly over the rectangular domain $[-1, 1] \times [-1, 1]$, the middle row and bottom row are the corresponding MCMC fitted mean surfaces and online SMC fitted mean surfaces at $T = 5000$ with $d = 6 \times 6$, *n.min* $= 40$.

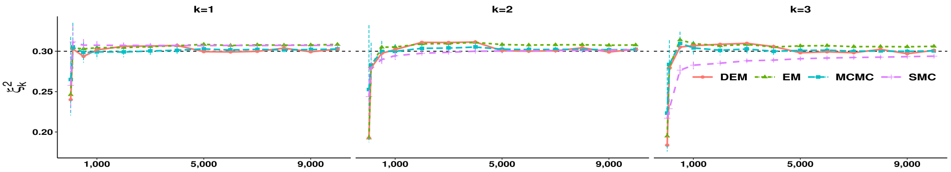

Figure A3. Estimated parameter of $\xi_k^2, k = 1, 2, 3$ with 95% equal tailed credible interval of online SMC algorithm versus MCMC algorithm (EM, DEM) when $K = 3$ and the number of observations increases from $t = 20$ to $10, 000$.

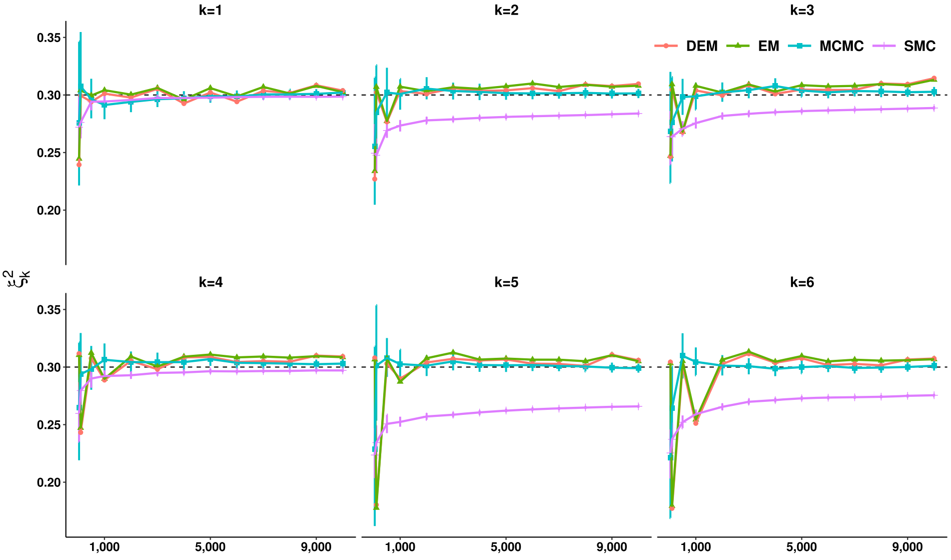

Figure A4. Estimated parameter $\xi_k^2, k = 1, \ldots, 6$ with 95% equal tailed credible interval of online SMC algorithm versus MCMC algorithm (EM, DEM) when $K = 6$ and the number of observations increases from $t = 40$ to $10, 000$.

Appendix 6. Real data analysis

Predictive information criteria

Predictive information criterion, such as expected predictive deviance (EPD), expected Akaike information criterion (EAIC), expected Bayesian information criterion (EBIC), Watanabe–Akaike information criterion (WAIC) are often used to measure model predictive accuracy in Bayesian framework [41,42]. In our proposed online SMC algorithm, we compute EPD, EAIC, EBIC, WAIC at time T after obtaining all the estimates of model parameters via Equations (A13)–(A16).

$$EPD = -2\sum_{t=1}^{T} \log E_{\theta}[p(\boldsymbol{y}_t|\theta)]. \tag{A13}$$

$$EAIC = -2\sum_{t=1}^{T} \log E_{\theta}[p(\boldsymbol{y}_t|\theta)] + 2v_{\boldsymbol{\theta}}. \tag{A14}$$

$$EBIC = -2\sum_{t=1}^{T} \log E_{\theta}[p(\boldsymbol{y}_t|\theta)] + v_{\boldsymbol{\theta}} \log(T). \tag{A15}$$

$$WAIC = -2\sum_{t=1}^{T} \log E_{\theta}[p(\boldsymbol{y}_t|\theta)] + 2\sum_{t=1}^{T} V(\log(p(\boldsymbol{y}_t|\boldsymbol{\theta}))). \tag{A16}$$

Here $v_{\boldsymbol{\theta}}$ in EAIC and EBIC denotes the number of parameters in the model, and in our proposed online SMC algorithm, we use $\sum_{i=1}^{N} p(\boldsymbol{y}_t|\boldsymbol{\theta}_T^{(i)})/N$ to approximate $E_{\boldsymbol{\theta}}[p(\boldsymbol{y}_t|\boldsymbol{\theta})]$, and $V(\log(p(\boldsymbol{y}_t|\boldsymbol{\theta})))$ is approximated via follows:

$$\frac{\sum_{i=1}^{N}(\log(p(\boldsymbol{y}_t|\boldsymbol{\theta}_T^{i})))^2 - \frac{(\sum_{i=1}^{N}\log(p(\boldsymbol{y}_t|\boldsymbol{\theta}_T^{(i)})))^2}{N}}{N-1},$$

and $p(\boldsymbol{y}_t|\boldsymbol{\theta}_T^{(i)})$ is estimated by $\sum_{k=1}^{K} p(\boldsymbol{y}_t|\boldsymbol{\theta}_T^{(i)}, z_t = k)p(z_t = k)$.

Handwritten number images

As indicated in Figure A5 and Figure A6, when $K = 8$, the MSSR$_\mathrm{m}$ model is able to capture the common features of the most of the 10 digits as well as subgroup of 0, except 3, 8, and as indicated by the 1*st* cluster, 7, 9 share some same common features. As indicated in Figure A6, when $K = 10$, the MSSR$_\mathrm{m}$ model is able to capture the common features for most of the 10 digits as well as subgroup of 0, 2, except 3, and 7, 9 (the 1*st* cluster) share some common features, as well as 8 and 2 (the 9*th* cluster).

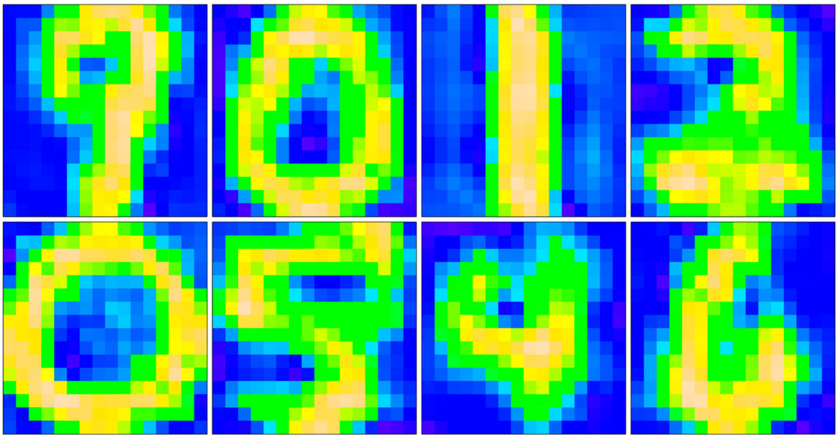

Figure A5. Online SMC estimated common features from MSSR$_{\mathrm{m}}$ model of handwritten number images by cluster when $K = 8$, $d = 8 \times 8$, labelled as the 1*st* cluster to the 8*th* cluster in left-to-right, top-to-bottom order.

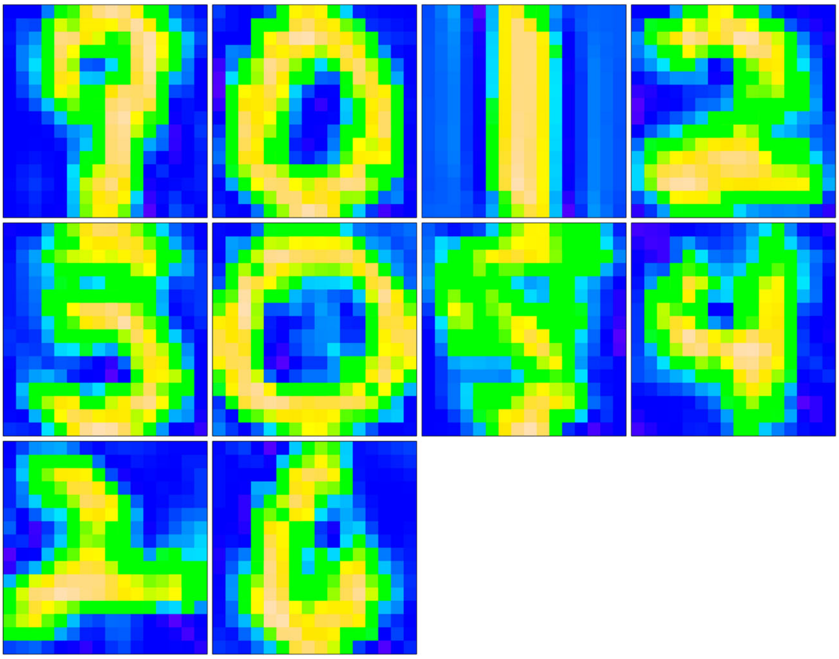

Figure A6. Online SMC estimated common features from MSSR$_{\mathrm{m}}$ model of handwritten number images by cluster when $K = 10$, $d = 8 \times 8$, labelled as the 1*st* cluster to the 10*th* cluster in left-to-right, top-to-bottom order.