Statistics and Its Interface Volume 15 (2022) 197–208

A sequential Monte Carlo Gibbs coupled with stochastically approximated expectation-maximization algorithm for functional data

Ziyue Liu

We develop an algorithm to overcome the curse of dimensionality in sequential Monte Carlo (SMC) for functional data. In the inner iterations of the algorithm for given parameter values, the conditional SMC is extended to obtain draws of the underlying state vectors. These draws in turn are used in the outer iterations to update the parameter values in the framework of stochastically approximated expectation-maximization to obtain maximum likelihood estimates of the parameters. Standard errors of the parameters are calculated using a stochastic approximation of Louis formula. Three numeric examples are used for illustration. They show that although the computational burden remains high, the algorithm produces reasonable results without exponentially increasing the particle numbers.

Keywords and phrases: Functional data, Gibbs sampler, Particle filter, Sequential Monte Carlo, State space model, Stochastically approximated EM.

1. INTRODUCTION

Nonlinear and non-Gaussian State space models (SSMs) are an extremely flexible framework in studying complex dynamics over time or other domains, and are widely used in many fields such as physics, engineering, economics, geoscience, and statistics. However, their applications to functional data remain limited. One major barrier to such applications is the curse of dimensionality known as particle collapse when approximating the analytically intractable integrals using sequential Monte Carlo (SMC, also known as particle filter). Particle collapse happens when majority but a few of the random draws have essentially zero weights, which renders the numeric approximations to fail. In this paper, we aim to develop an algorithm applicable to functional data which does not suffer from particle collapse.

Linear Gaussian SSMs have been applied to functional data. [1, 2] represented the functional mixed effects models (FMMs) in state space form and used it for fast computations. In these models, both the population-level and the subject-level functional effects were modeled by smoothing splines, whose state space forms are readily available. [3, 4] used state space models directly as the building blocks for functional data which can incorporate more dynamic structures beyond nonparametric curves. [5] proposed to use stochastic processes as the building blocks, which in principle can be casted as SSMs when they are Markovian. On the other hand, nonlinearity or non-Gaussianity or both are needed in many cases [6, 7]. For such situations, nonlinear and non-Gaussian SSMs would provide a flexible and unified framework for estimation, inference and prediction. Since exact algorithms no longer exist, numeric approximations will be needed for implementations. SMC is the most popular approach in recent decades, which has been applied to single-subject problems [8, 9].

SMC is a set of sequential importance-sampling resampling algorithms [10, 11]. At each time points, a number of random draws (known as particles) are generated with importance weights. Functions of the underlying states including the likelihood are approximated by these particles. The performance of the approximations is largely decided by the numbers of particles and the uniformity of the weights, which is summarized by the effective sample size (ESS) [12, 13]. ESS is calculated as the inverse of the sum of the squared normalized weights. Resampling is recommended when ESS drops below a certain threshold, for example two-thirds of the number of particles. This approach works when the state dimension is low that ESS only decreases gradually as systems move forward in time. However, when the state dimension is high, it has been observed that the weights of majority but a few particles are numerically zero even after one step moving forward [14]. This curse of dimensionality is known as particle collapse, which causes the numeric approximations to fail. Particle collapse cannot be overcome by resampling because there are not enough valid particles to resample from.

[15, 16] studied the theoretical properties of particle collapse. They proved that in order to avoid particle collapse, the number of particles must be exponential in the state dimensions. [17] further established that the number of particles needs to scale exponentially with the Kullback-Leibler divergence between the target and the proposal densities. This requirement of exponential increase is especially unfriendly to functional data, because even for a fixed model

structure, the state dimensions increase linearly with more subjects.

In literature, several methods have been developed to overcome particle collapse without exponentially increasing the numbers of particles. [18, 19] used separate filters for blocks of the whole state vectors by assuming working independence structures, which targeted the marginal distributions of the blocks. [20] proved that in principle it is possible to develop algorithms with dimension-free approximation errors when the dependency structure is local at each time point. This approach is inapplicable to functional data because the population-level effects will introduce global dependency through observations. [21] proposed a blocked conditional particle filter, where the state vectors are divided into fully conditional blocks for filtering and smoothing. Their algorithms still require the calculation of the full state transition density and the system parameters are modeled with priors in the Bayesian framework. [22] proposed a state-time particle filter, where at each time point a local filter is adopted along the space direction for each particle island and the information in the data is brought in gradually. They investigated both theoretical and numeric properties for a linear Gaussian SSM and a finite-state SSM, for both of which exact filtering and smoothing algorithms exist.

In this paper, we propose a new algorithm applicable to functional data with SMC Gibbs sampler nested in the stochastically approximated expectation-maximization algorithm [23, 24]. It is straightforward to observe that given the subject-specific effects, the population-level components have fixed low dimensions, while given the later, the former ones are independently related to the data. Therefore, we extend the conditional SMC to obtain draws of the state vectors for given parameter values [25]. The working state vector at each SMC Gibbs step has fixed low dimensions, which precludes particle collapse. These draws are used to approximate the conditional expectation of the logarithm of the complete data likelihood (the Q function). The Q function is then maximized to update the parameter values. Maximum likelihood estimates (MLEs) of the parameters are obtained after the algorithm converges. Their variances can be estimated using Louis formula [26]. The state vectors are estimated by their conditional means through averaging the corresponding conditional draws. Functional effects are then calculated from the state estimates.

The remaining of this paper is organized as follows. Section 2 presents the nonlinear non-Gaussian SSMs for functional data. Section 3 presents the proposed algorithm. Section 4 illustrates the proposed algorithm using several numeric examples. Some discussions are given in Section 5.

2. THE MODEL

Let y_{ij} denote the observation from the ith subject at time t_j with $i = 1, \cdots, n$ and $j = 1, \cdots, T_i$. For the simplicity of notations, assume that $T_i = T$ and all subjects are observed on the same time grid, although these assumptions are not necessary for the proposed algorithm. Let $\mathrm{E}\{\cdot\}$ denote the expectation with respect to an appropriate distribution, and $g_y(\cdot)$ the link function. For example, $g_y(\cdot)$ can be the logit link for binary outcomes, logarithm link for count outcomes, or the identity link for Gaussian outcomes. Let $\boldsymbol{\beta}(t_j)$ be the population-level effect vector, $\boldsymbol{\alpha}_i(t_j)$ the subject-specific effect vector, and X_{ij} and Z_{ij} the corresponding design matrices formulated by incorporating the study designs and covariates. Let $\boldsymbol{\theta}$ be the vector of the unknown but fixed parameters. The *data observation equation* is

$$(1)\quad g_y\left[\mathrm{E}\left\{y_{ij} \mid \boldsymbol{\beta}(t_j), \boldsymbol{\alpha}_i(t_j); \boldsymbol{\theta}\right\}\right] = X_{ij}\boldsymbol{\beta}(t_j) + Z_{ij}\boldsymbol{\alpha}_i(t_j).$$

The *state observation equations* define how $\boldsymbol{\beta}(t_j)$ and $\boldsymbol{\alpha}_i(t_j)$ are transformed from their corresponding latent states, $\boldsymbol{\phi}(t_j)$ and $\boldsymbol{\psi}_i(t_j)$, through observational matrices F_β and F_α as follows

$$(2)\quad \boldsymbol{\beta}(t_j) = F_\beta \boldsymbol{\phi}(t_j),$$
$$(3)\quad \boldsymbol{\alpha}_i(t_j) = F_\alpha \boldsymbol{\psi}_i(t_j).$$

The state vectors $\boldsymbol{\phi}(t_j)$ and $\boldsymbol{\psi}_i(t_j)$ for $i = 1, \cdots, n$ are mutually independent. Their evolutions over time are defined through the two conditional probability density functions (PDFs), $p_\phi\{\cdot \mid \cdot\}$ and $p_\psi\{\cdot \mid \cdot\}$ where $p\{\cdot\}$ stand for a PDF in general, as in the following *state transition equations*

$$(4)\quad \boldsymbol{\phi}(t_j) \sim p_\phi\left\{\boldsymbol{\phi}(t_j) \mid \boldsymbol{\phi}(t_{j-1}); \boldsymbol{\theta}\right\},$$
$$(5)\quad \boldsymbol{\psi}_i(t_j) \sim p_\psi\left\{\boldsymbol{\psi}_i(t_j) \mid \boldsymbol{\psi}_i(t_{j-1}); \boldsymbol{\theta}\right\},$$

The system is initialized at time zero according to certain distributions $p_{\phi 0}\{\boldsymbol{\phi}(0)\}$ and $p_{\psi 0}\{\boldsymbol{\psi}_i(0)\}$. Equations (4) and (5) are not limited to first order Markov processes. Higher order processes can be incorporated by expanding the state vectors.

Equations (1) to (5) define how to model functional data using state space models. It extends [4] to nonlinear and non-Gaussian models. Besides nonparametric curves such as smoothing splines [27, 1], many other dynamics over time such as parametric fixed and random effects [28], multiprocess models [29, 30], and differential equation based models [31, 7] can be incorporated. More examples of SSMs can be found in [32, 33]. Consequently, the resultant models are more flexible than the traditional nonparametric curve based FMMs.

Estimation for these models is twofold: estimating the latent state vectors for given parameter values, from which $\boldsymbol{\beta}(t_j)$ and $\boldsymbol{\alpha}_i(t_j)$ will be calculated, and estimating the unknown but fixed parameters. For the first task, SMC will be adopted. The conditional SMC update will be extended to overcome particle collapse. For the second task, EM algorithm is an ideal framework because of two reasons. First, given the state vectors, the likelihood is straightforward to calculate and maximize, for which close-form solutions exist

in many cases. Second, the SMC approximated likelihood functions are not continuous with respect to parameters, which causes serious problems for gradient based maximization methods [34]. Although [34] developed an algorithm for continuous SMC likelihood approximation for univariate state situations, a general solution for multivariate state vectors remains unavailable.

3. THE ALGORITHM

For the simplicity of notations, let '$\top$' denote matrix transpose, and $\boldsymbol{\beta}_j = \boldsymbol{\beta}(t_j)$, $\boldsymbol{\alpha}_{ij} = \boldsymbol{\alpha}_i(t_j)$, $\boldsymbol{\phi}_j = \boldsymbol{\phi}(t_j)$, $\boldsymbol{\psi}_{ij} = \boldsymbol{\psi}_i(t_j)$,

$$\begin{aligned}
\boldsymbol{y}_{i,1:T} &= \begin{pmatrix} y_{i1} & \cdots & y_{iT} \end{pmatrix}^\top, \\
\boldsymbol{y}_j &= \begin{pmatrix} y_{1j} & \cdots & y_{nj} \end{pmatrix}^\top, \\
\boldsymbol{y} &= \begin{pmatrix} \boldsymbol{y}_1^\top & \cdots & \boldsymbol{y}_T^\top \end{pmatrix}^\top, \\
\boldsymbol{\phi} &= \begin{pmatrix} \boldsymbol{\phi}_1^\top & \cdots & \boldsymbol{\phi}_T^\top \end{pmatrix}^\top, \\
\boldsymbol{\psi}_{i,1:T} &= \begin{pmatrix} \boldsymbol{\psi}_{i1}^\top & \cdots \boldsymbol{\psi}_{iT}^\top \end{pmatrix}^\top \\
\boldsymbol{\psi}_j &= \begin{pmatrix} \boldsymbol{\psi}_{1j}^\top & \cdots & \boldsymbol{\psi}_{nj}^\top \end{pmatrix}^\top, \\
\boldsymbol{\psi} &= \begin{pmatrix} \boldsymbol{\psi}_1^\top & \cdots & \boldsymbol{\psi}_T^\top \end{pmatrix}^\top, \\
\boldsymbol{\xi}_j &= \begin{pmatrix} \boldsymbol{\phi}_j^\top & \boldsymbol{\psi}_j^\top \end{pmatrix}^\top, \\
\boldsymbol{\xi} &= \begin{pmatrix} \boldsymbol{\xi}_1^\top & \cdots & \boldsymbol{\xi}_T^\top \end{pmatrix}^\top.
\end{aligned}$$

We first briefly review SAEM and particle Gibbs, and then introduce the proposed algorithm.

3.1 SAEM

The SAEM algorithm developed by [23] can be summarized as follows. First define the Q function

$$Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}'\right) = \int \log\left\{p\left(\boldsymbol{y},\boldsymbol{\phi},\boldsymbol{\psi}|\boldsymbol{\theta}\right)\right\} p\left(\boldsymbol{\phi},\boldsymbol{\psi}|\boldsymbol{y},\boldsymbol{\theta}'\right)\, d\boldsymbol{\phi}\, d\boldsymbol{\psi}. \tag{6}$$

In the regular EM algorithms, the E-step at iteration k is to calculate Q_k as a function of $\boldsymbol{\theta}$

$$Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}_k\right) = \int \log\left\{p\left(\boldsymbol{y},\boldsymbol{\phi},\boldsymbol{\psi}|\boldsymbol{\theta}\right)\right\} p\left(\boldsymbol{\phi},\boldsymbol{\psi}|\boldsymbol{y},\boldsymbol{\theta}_k\right)\, d\boldsymbol{\phi}\, d\boldsymbol{\psi}. \tag{7}$$

The M-step maximizes Q_k that updates the parameter estimate to $\boldsymbol{\theta}_{k+1}$. When the E-step is analytically intractable, the Q function is approximated by $m(k)$ realizations of $\{\boldsymbol{\phi}(l), \boldsymbol{\psi}(l)\}$ from $p\left(\boldsymbol{\phi},\boldsymbol{\psi}|\boldsymbol{y},\boldsymbol{\theta}_k\right)$ as

$$\hat{Q}_k = (1-\gamma_k)\,\hat{Q}_{k-1} + \gamma_k \sum_{l=1}^{m(k)} \log\left\{p\left(\boldsymbol{y},\boldsymbol{\phi}(l),\boldsymbol{\psi}(l)|\boldsymbol{\theta}\right)\right\}. \tag{8}$$

γ_k's are step sizes that satisfy (i) $0 \le \gamma_k \le 1$, (ii) $\sum_{k=1}^{\infty}\gamma_k = \infty$, and (iii) $\sum_{k=1}^{\infty}\gamma_k^2 < \infty$.

For nonlinear and non-Gaussian SSMs, realizations of $\{\boldsymbol{\phi}(l), \boldsymbol{\psi}(l)\}$ from $p\left(\boldsymbol{\phi},\boldsymbol{\psi}|\boldsymbol{y},\boldsymbol{\theta}_k\right)$ have to be approximated numerically, for example using SMC.

3.2 Particle Gibbs

A generic SMC algorithm [11] sequentially approximates $p(\boldsymbol{\xi}_j|\boldsymbol{y}_1, \cdots, \boldsymbol{y}_j;\boldsymbol{\theta})$ by N particles $\boldsymbol{\xi}_j^{1:N}$ and their importance weight $v_j^{1:N}$ as follows

1: at time $j = 1$, for particle $l = 1, \cdots, N$,
   (a) sample from proposal $\boldsymbol{\xi}_1^l \sim q\left(\cdot|\boldsymbol{y}_1;\boldsymbol{\theta}\right)$
   (b) calculate the weights

$$w_1^l = \frac{p\left(\boldsymbol{y}_1|\boldsymbol{\xi}_1^l;\boldsymbol{\theta}\right) p\left(\boldsymbol{\xi}_1^l;\boldsymbol{\theta}\right)}{q\left(\boldsymbol{\xi}_1^l|\boldsymbol{y}_1;\boldsymbol{\theta}\right)} \tag{9}$$

   (c) normalize the weights $v_1^l = w_1^l / \sum_{m=1}^N w_1^m$. The collection of these normalized weight $v_j^{1:N}$ formulate a discrete probability distribution denoted as $R(\cdot|v_j^{1:N})$.

2: at time $j = 2, \cdots, T$, for particles $l = 1, \cdots, N$
   (a) resample by selecting ancestor $a_j^l \sim R\left(\cdot|v_{j-1}^{1:N}\right)$
   (b) moving the system one step forward from proposal $\boldsymbol{\xi}_j \sim q(\cdot|\boldsymbol{\xi}_{j-1}^{a_j^l}, \boldsymbol{y}_j;\boldsymbol{\theta})$
   (c) calculate the weights

$$w_j^l = \frac{p\left(\boldsymbol{y}_j|\boldsymbol{\xi}_j^l;\boldsymbol{\theta}\right) p\left(\boldsymbol{\xi}_j^l|\boldsymbol{\xi}_{j-1}^{a_j^l};\boldsymbol{\theta}\right)}{q\left(\boldsymbol{\xi}_j^l|\boldsymbol{\xi}_{j-1}^{a_j^l}, \boldsymbol{y}_j;\boldsymbol{\theta}\right)} \tag{10}$$

   (d) normalize the weights $v_j^l = w_j^l / \sum_{m=1}^N w_j^m$ to formulate $R(\cdot|v_j^{1:N})$.

3: approximate functions of $\boldsymbol{\xi}_j$ by $\boldsymbol{\xi}_j^{1:N}$ and $v_j^{1:N}$, for example, the likelihood function.

The output are the collections of particles $\boldsymbol{\xi}_{1:T}^{1:N} = (\boldsymbol{\xi}_1^{1:N}, \cdots, \boldsymbol{\xi}_T^{1:N})$, their corresponding weights $v_{1:T}^{1:N}$ and their ancestry $a_{2:T}^{1:N}$. In the original SMC [10], the state transition equation was used to move the system forward, and bootstrap was used for resampling. Some developments after [10] were summarized in the review papers [35, 36, 37].

Resampling is not necessary at all steps, though it is recommended when ESS drops below a certain threshold such as $2N/3$. ESS is calculated as

$$\text{ESS} = \frac{1}{\sum_{l=1}^N \left(v_j^l\right)^2}. \tag{11}$$

For low dimension problems, ESS decreases gradually over time. When the state dimensions are high, majority except a few of $v_j^{1:N}$ could be be zero even after one step moving forward. Consequently, the particle approximation of $p(\boldsymbol{\xi}_j|\boldsymbol{y}_1, \cdots, \boldsymbol{y}_j;\boldsymbol{\theta})$ fails. To avoid this particle collapse, N is required to scale exponentially with the state dimensions [15, 16].

The conditional SMC update proposed by [25] can be outlined as follows.

1: let $\boldsymbol{\xi}^*_{1:T} = (\boldsymbol{\xi}^*_1, \cdots, \boldsymbol{\xi}^*_T)$ be a prespecified sequence.
2: at time $j = 1$
   (a) let $\boldsymbol{\xi}^1_1 = \boldsymbol{\xi}^*_1$
   (b) for $l = 2, \cdots, N$, sample from $\boldsymbol{\xi}^l_1 \sim q(\cdot|\boldsymbol{y}_1; \boldsymbol{\theta})$
   (c) for $l = 1, \cdots, N$, calculate weights as in (9) and normalize weights to obtain $v^{1:N}_1$.
3: at time $j = 2, \cdots, T$
   (a) let $\boldsymbol{\xi}^1_j = \boldsymbol{\xi}^*_j$
   (b) for $l = 2, \cdots, N$,
      (i) resample by selecting ancestor $a^l_j \sim R(\cdot|v^{1:N}_{j-1})$
      (ii) move the system one step forward from proposal $\boldsymbol{\xi}_j \sim q(\cdot|\boldsymbol{\xi}^{a^l_j}_{j-1}, \boldsymbol{y}_j; \boldsymbol{\theta})$
   (c) for $l = 1, \cdots, N$, calculate weights as in (10) and normalize weights to obtain $v^{1:N}_j$.

This conditional SMC update is constructed on an extended distribution including both the draws and their ancestry. Consequently, the marginal of the output is the target distribution $p(\boldsymbol{\xi}|\boldsymbol{y}; \boldsymbol{\theta})$ by retaining relevant variables.

Particle Gibbs is a Markov chain Monte Carlo (MCMC) method under the Bayesian framework outlined as follows.

1: set the prior distribution for $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$.
2: initialize at $k = 0$ by setting $\boldsymbol{\theta}(0)$ and $\boldsymbol{\xi}^*_{1:T}(0)$ to arbitrary values.
3: for iterations $k \geq 1$
   (a) sample $\boldsymbol{\theta}(k) \sim p(\cdot|\boldsymbol{y}, \boldsymbol{\xi}^*_{1:T}(k-1))$
   (b) run the conditional SMC update conditional on $\boldsymbol{\xi}^*_{1:T}(k-1)$
   (c) sample $\boldsymbol{\xi}^*_{1:T}(j)$ from the output of the conditional SMC update.

3.3 The proposed algorithm

We adopt a frequentist approach in stead of Bayesian by using SAEM for parameter estimation. We propose to obtain draws of $\{\boldsymbol{\phi}(l), \boldsymbol{\psi}(l)\}$ from $p(\boldsymbol{\phi}, \boldsymbol{\psi}|\boldsymbol{y}, \boldsymbol{\theta}_k)$ by extending the conditional SMC update to avoid particle collapse without exponentially increasing N.

It is straightforward to observe that given $\boldsymbol{\phi}$, we have n independent models for $\boldsymbol{y}_{i,1:T}$'s defined by

$$\boldsymbol{y}_{i,1:T} \sim p\left(\boldsymbol{y}_{i,1:T}|\boldsymbol{\psi}_i; \boldsymbol{\theta}, \boldsymbol{\phi}\right) p\left(\boldsymbol{\psi}_i; \boldsymbol{\theta}\right), \tag{12}$$

where $\boldsymbol{\phi}$ should be viewed as a constant vector similar to $\boldsymbol{\theta}$ instead of a random vector. This suggests that separate SMC can be performed for the n subjects to draw $\boldsymbol{\psi}_{i,1:T}$. For each subject, the working state vectors are of fixed and low dimension. On the other hand, when $\boldsymbol{\psi}_{i,1:T}$, $i = 1, \cdots, n$ are given, we have n observations at each time point as

$$y_{ij} \sim p\left(y_{ij}|\boldsymbol{\phi}_j; \boldsymbol{\theta}, \boldsymbol{\psi}_{ij}\right) p\left(\boldsymbol{\phi}_j; \boldsymbol{\theta}\right), \tag{13}$$

where $\boldsymbol{\psi}_{ij}$ should be viewed as constant. In this situation, $\boldsymbol{\phi}_j$, though observed n times, is of low dimension that does not depend on n. Consequently, a Gibbs iteration can be adopted to obtain joint draw of $(\boldsymbol{\phi}, \boldsymbol{\psi})$. To ensure that $p(\boldsymbol{\phi}, \boldsymbol{\psi}|\boldsymbol{y}; \boldsymbol{\theta})$ is the targeting distribution, the conditional SMC is adopted.

We first modify the conditional SMC update to draw from (12) and (13). For $(\boldsymbol{\phi}|\boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{\psi})$, the modified conditional SMC update is as follows

1: let $\boldsymbol{\phi}^* = (\boldsymbol{\phi}^*_1, \cdots, \boldsymbol{\phi}^*_T)$ be a prespecified sequence and $\boldsymbol{\psi}^*$ be known values.
2: at time $j = 1$
   (a) let $\boldsymbol{\phi}^1_1 = \boldsymbol{\phi}^*_1$
   (b) for particles $l = 2, \cdots, N$, sample from $\boldsymbol{\phi}_1 \sim q(\cdot|\boldsymbol{y}_1; \boldsymbol{\theta}, \boldsymbol{\psi}^*)$
   (c) for $l = 1, \cdots, N$, calculate weights as

$$w^l_1 = \frac{p\left(\boldsymbol{y}_1|\boldsymbol{\phi}^l_1; \boldsymbol{\theta}, \boldsymbol{\psi}^*\right) p\left(\boldsymbol{\phi}^l_1; \boldsymbol{\theta}\right)}{q\left(\boldsymbol{\phi}^l_1|\boldsymbol{y}_1; \boldsymbol{\theta}, \boldsymbol{\psi}^*\right)} \tag{14}$$

   (d) normalize the weight $v^l_1 = w^l_1 / \sum_{m=1}^N w^m_1$.
3: at time $j = 2, \cdots, T$
   (a) let $\boldsymbol{\phi}^1_j = \boldsymbol{\phi}^*_j$
   (b) for particles $l = 2, \cdots, N$
      (i) resample by selecting ancestor $a^l_j \sim R\left(\cdot|v^{1:N}_{j-1}\right)$
      (ii) move the system one step forward from proposal $\boldsymbol{\phi}_j \sim q(\cdot|\boldsymbol{y}_j, \boldsymbol{\phi}^{a^l_j}_{j-1}; \boldsymbol{\theta}, \boldsymbol{\psi}^*)$
   (c) for $l = 1, \cdots, N$, calculate weights as

$$w^l_j = \frac{p\left(\boldsymbol{y}_j|\boldsymbol{\phi}^l_j; \boldsymbol{\theta}, \boldsymbol{\psi}^*\right) p\left(\boldsymbol{\phi}^l_j|\boldsymbol{\phi}^{a^l_j}_{j-1}; \boldsymbol{\theta}\right)}{q\left(\boldsymbol{\phi}^l_j|\boldsymbol{\phi}^{a^l_j}_{j-1}, \boldsymbol{y}_j; \boldsymbol{\theta}, \boldsymbol{\psi}^*\right)} \tag{15}$$

   (d) normalize the weight $v^l_j = w^l_j / \sum_{m=1}^N w^m_j$.
4: sample $\boldsymbol{\phi}^*$ from the output of the above conditional SMC update.

The conditional SMC update for $(\boldsymbol{\psi}_{i,1:T}|\boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{\phi})$ follows similarly, except that (i) the update is performed separately for subject $i = 1, \cdots, n$, and (ii) at each time, data y_{ij} instead of $\boldsymbol{y}_j$ is used. The product of the ith update is $\boldsymbol{\psi}^*_{i,1:T}$. The collection of $\boldsymbol{\psi}^*_{i,1:T}$'s are the updated $\boldsymbol{\psi}^*$.

The proposed algorithm is outlined as follows.

1: initialize at $k = 0$
   (a) set $\boldsymbol{\theta}(0)$, $\boldsymbol{\phi}^*(0)$, and $\boldsymbol{\psi}^*(0)$ to arbitrary values
   (b) set $\hat{Q}_0 = 0$ (or an arbitrary value)
   (c) set the step values γ_k's, where $\gamma_1 = 1$ so that the artificial information in $\hat{Q}_0$ will not be incorporated for MLEs.

2: for outer iterations $k \geq 1$

(a) let $\boldsymbol{\phi}^*(0,k) = \boldsymbol{\phi}^*(k-1)$, and $\boldsymbol{\psi}^*(0,k) = \boldsymbol{\psi}^*(k-1)$

(b) for inner iteration $q \geq 1$

(i) run the modified conditional SMC update on $\boldsymbol{\phi}$ conditional on $\boldsymbol{\psi}^*(q-1,k)$ and $\boldsymbol{\phi}^*(q-1,k)$ to obtain $\boldsymbol{\phi}^*(q,k)$

(ii) for subject $i = 1, \cdots, n$, run the modified conditional SMC update on $\boldsymbol{\psi}_{i,1:T}$ conditional on $\boldsymbol{\phi}^*(q,k)$ and $\boldsymbol{\psi}^*(q-1,k)$ to obtain $\boldsymbol{\psi}^*_{i,1:T}(q,k)$, the collection of which formulate $\boldsymbol{\psi}^*(q,k)$

(c) assume that the inner iterations converge after n_q iterations, then let $\boldsymbol{\phi}^*(k) = \boldsymbol{\phi}^*(n_q,k)$ and $\boldsymbol{\psi}^*(k) = \boldsymbol{\psi}^*(n_q,k)$

(d) run $m(k)$ chains to obtain $m(k)$ copies of $\boldsymbol{\phi}^*(k)$ and $\boldsymbol{\psi}^*(k)$

(e) calculate $\hat{Q}_k$ according to (8)

(f) maximize $\hat{Q}_k$ to update the parameter estimate to $\boldsymbol{\theta}_k$.

[23] showed that convergence is guaranteed even $m(k) = 1$. Since "particle Gibbs" is used for the Bayesian framework [25] as reviewed in the previous subsection, we call the inner iterations "SMC Gibbs" to avoid confusion.

MLEs of the parameters are obtained after convergence. Convergence of the inner iterations can be assessed by methods used in MCMC monitoring such as trace plots [38]. Convergence of the outer iterations can be assessed similarly [39]. After convergence, the state vectors can be estimated by generating a number of conditional draws, from which the conditional means, variances, and confidence intervals of $\boldsymbol{\beta}_j$ and $\boldsymbol{\alpha}_{ij}$ can be calculated.

3.4 Standard errors

Let $I_o(\boldsymbol{\theta})$, $I_c(\boldsymbol{\theta})$, $I_m(\boldsymbol{\theta})$ be the observed, the complete, and the missing information matrices, respectively. Let l_c denote the logarithm of the complete data likelihood. The asymptotic variances of the parameters can be obtained using Louis formula [26] as follow

$$I_o(\boldsymbol{\theta}) = I_c(\boldsymbol{\theta}) - I_m(\boldsymbol{\theta}), \tag{16}$$

$$I_c(\boldsymbol{\theta}) = \mathrm{E}\left(-\frac{\partial^2 l_c}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}|\boldsymbol{y}\right), \tag{17}$$

$$I_m(\boldsymbol{\theta}) = \mathrm{E}\left(\frac{\partial l_c}{\partial\boldsymbol{\theta}}\frac{\partial l_c}{\partial\boldsymbol{\theta}^\top}|\boldsymbol{y}\right) - \mathrm{E}\left(\frac{\partial l_c}{\partial\boldsymbol{\theta}}\right)\mathrm{E}\left(\frac{\partial l_c}{\partial\boldsymbol{\theta}^\top}|\boldsymbol{y}\right). \tag{18}$$

The second term in Equation (18) is zero at MLE. [23] suggests to stochastically approximate the observed information matrix from Louis formula by H_k as

$$\begin{aligned}
H_k &= G_k - \Delta_k\Delta_k^\top,\\
\Delta_k &= (1-\gamma_k)\Delta_{k-1} + \gamma_k\frac{1}{m_k}\sum_{j=1}^{m_k}\partial_{\boldsymbol{\theta}} l_c(\boldsymbol{\theta}_k),\\
G_k &= (1-\gamma_k)G_{k-1}\\
&\quad + \gamma_k\frac{1}{m_k}\sum_{j=1}^{m_k}\left[-\partial^2_{\boldsymbol{\theta}} l_c(\boldsymbol{\theta}_k) + \partial_{\boldsymbol{\theta}} l_c(\boldsymbol{\theta}_k)\,\partial_{\boldsymbol{\theta}^\top} l_c(\boldsymbol{\theta}_k)\right],
\end{aligned}$$

where G_k and Δ_k can be initialized at arbitrary values. They showed that H_k converges to the observed information matrix evaluated at $\boldsymbol{\theta}^*$, where $\boldsymbol{\theta}^*$ is a limiting point of SAEM. Inverse of the observation matrix gives the asymptotic variance matrix $\mathrm{var}(\boldsymbol{\theta}) = I_o^{-1}(\boldsymbol{\theta})$. Square roots of the diagonal elements of $\mathrm{var}(\boldsymbol{\theta})$ give the standard errors (SEs).

3.5 Special cases of the proposed algorithm

Exact algorithms exist for linear Gaussian SSMs [32, 33], hence SMC is not needed. However, the Gibbs sampling approach could be useful when the state dimensions become extremely high. Since the exact algorithms involve multiplication and inversion of matrices scaling with the state dimensions, numeric difficulties and inaccuracies may happen for high dimension problems. We will illustrate how to apply a simplified version of the proposed algorithm to linear Gaussian models by removing the SMC components.

When the population-level effects can be adequately characterized by parametric fixed effects, these effects then become part of the parameter vector $\boldsymbol{\theta}$. Consequently, the models for the n subjects are independent. The proposed algorithm can be simplified by removing the inner iterations and directly applying n separate conditional SMC updates to obtain conditional draws for $\boldsymbol{\psi}_{i,1:T}$. Such coupling of particle MCMC with EM for nonlinear non-Gaussian SSMs have been studied for the single time series situations in literature [40, 41, 42].

4. NUMERIC EXAMPLES

In this section, numeric examples are used to illustrate the proposed algorithm, with comparisons with the multiple particle filter (MPF) [18] and the space-time particle filter (STPF) [22]. Since neither smoothing algorithms nor parameter estimation procedures are given for MPF and STPF, the filtered means are used for comparisons based on the estimate parameters from the proposed algorithm.

A linear Gaussian model and a Poisson outcome model using cubic smoothing spline based FMMs [1] are used. The identity link is used for the linear Gaussian model and the logarithm link is used for the Poisson outcome. Both models have the same state observation equations (2) and (3) with $F_\beta = F_\alpha = \begin{pmatrix} 1 & 0 \end{pmatrix}$. Let $\Delta t = t_j - t_{j-1}$, H the state transition matrix, $\boldsymbol{\eta}_{\phi j} \sim \mathrm{N}(\mathbf{0}, \lambda_1\Sigma)$ and $\boldsymbol{\eta}_{\psi ij} \sim \mathrm{N}(\mathbf{0}, \lambda_2\Sigma)$ the state innovation vectors with

$$H = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \frac{1}{3}(\Delta t)^3 & \frac{1}{2}(\Delta t)^2 \\ \frac{1}{2}(\Delta t)^2 & \Delta t \end{pmatrix}.$$

The state transition equations are

$$\boldsymbol{\phi}_j = H\boldsymbol{\phi}_{j-1} + \boldsymbol{\eta}_{\phi j}, \quad \boldsymbol{\psi}_{ij} = H\boldsymbol{\psi}_{i,j-1} + \boldsymbol{\eta}_{\psi ij}, \tag{19}$$

At time zero, $\boldsymbol{\phi}_0 \sim \mathrm{N}(0, \kappa I_2)$ with $\kappa \to \infty$ and I_2 is the 2×2 identity matrix, and $\boldsymbol{\psi}_{i0} \sim \mathrm{N}(0, \text{diagonal}\{\sigma_1^2, \sigma_2^2\})$.

For these numeric examples, we let $m(k) = 1$ for all k's and the number of particles $N = 1000$. The classical bootstrap filter is used for filtering [10], which uses the state transition equations as the proposal densities and multinomial resampling with replacement to select ancestors. The algorithm proposed by [44] is used to sample $\boldsymbol{\phi}^*(q,k)$ and $\boldsymbol{\psi}^*(q,k)$ from the conditional SMC update outputs. In the follows we illustrate how to use [44] to draw one series for $\boldsymbol{\phi}(k)$ from the bootstrap filter output.

1: at outer iteration k and inner iteration q, let $\boldsymbol{\phi}_{1:T}^{1:N}$ and $v_{1:T}^{1:n}$ be the output from the modified conditional SMC update.
2: at time T, sample $\boldsymbol{\phi}_T^*$ from $\boldsymbol{\phi}_T^{1:N}$ with weights $v_T^{1:N}$.
3: for $j = T-1, \cdots, 1$
   (a) for $l = 1, \cdots, N$, calculate weights

$$\zeta_j^l = p\left(\boldsymbol{\phi}_{j+1}^* | \boldsymbol{\phi}_j^l; \boldsymbol{\theta}(k)\right) \tag{20}$$

   (b) normalize weights as $\omega_j^l = \zeta_j^l / \sum_{m=1}^N \zeta_j^m$
   (c) sample $\boldsymbol{\phi}_j^*$ from $\boldsymbol{\phi}_j^{1:N}$ according to weights $\omega_j^{1:N}$.
4: $\boldsymbol{\phi}^*(q,k) = (\boldsymbol{\phi}_1^*, \cdots, \boldsymbol{\phi}_T^*)$ is updated.

For both models, the M-step is straightforward. A brief summary of the M-step is given in Appendix.

4.1 Linear Gaussian model

The data were generated from a linear Gaussian model as follows

$$y_{ij} = \beta(t_j) + \alpha_i(t_j) + \varepsilon_{ij}, \tag{21}$$

for $i = 1, \cdots, 30$ subjects and $j = 1, \cdots, 50$ equally spaced time points on $[0, 1]$. Errors ε_{ij}'s were independent and identically distributed as the standard normal $\mathrm{N}(0, \sigma_e^2 = 1)$. The parameter vector is $\boldsymbol{\theta} = (\ \lambda_1 \quad \sigma_1^2 \quad \sigma_2^2 \quad \lambda_2 \quad \sigma_e^2\ )^\top$. Let $\boldsymbol{t} = (\ t_1 \quad \cdots t_T\ )^\top$, $\beta(\boldsymbol{t})$ and $\alpha_i(\boldsymbol{t})$ were generated by first generating curves from $\mathrm{N}(0, \kappa\Omega(\boldsymbol{t}))$ where $\kappa > 0$ and $\Omega(\boldsymbol{t}) = R^1(\boldsymbol{t}, \boldsymbol{t})$ defined recursively using the Bernoulli polynomial as [43]

$$k_1(t) = t - 0.5, \tag{22}$$

$$k_2(t) = \frac{1}{2}\left\{k_1^2(t) - \frac{1}{12}\right\}. \tag{23}$$

$$k_4(t) = \frac{1}{24}\left\{k_1^4(t) - \frac{1}{2}k_1^2(t) + \frac{7}{240}\right\}, \tag{24}$$

$$R^1(t_1, t_2) = k_2(t_1)\, k_2(t_2) - k_4(t_1 - t_2). \tag{25}$$

The curves were then rescaled and centered to obtain $\beta(t)$ and $\alpha_i(t)$ as displayed in Figure 1.

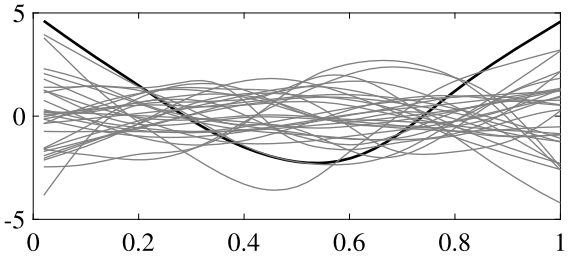

Figure 1. True functional effects: the thick black line is $\beta(t)$ and the thin gray lines are $\alpha_i(t)$'s.

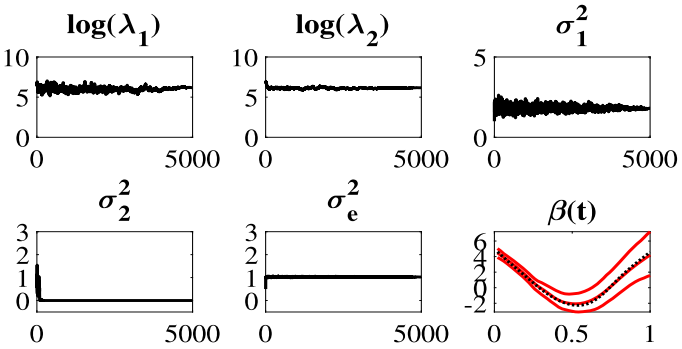

Figure 2. Case 1: Trace plots of parameters and the estimate of $\beta(t)$. For $\beta(t)$, the dotted black line is the true function and the solid red lines are the estimate and point-wise 95% confidence interval.

4.1.1 Case 1: SAEM with exact Gibbs

For linear Gaussian SSMs, exact algorithms are available to directly draw state vectors conditional on data when the state dimensions are low to moderate [45, 46]. When the state dimensions become very high, for example for functional data, the proposed algorithm can be modified by replacing the conditional SMC updates with the exact draws in the inner iterations [45, 46]. For the data example, $n_k = 5000$ outer iterations and $n_q = 100$ inner iterations were carried out. The trace plots of the parameters are displayed in Figure 2. It suggests that the algorithm has converged. $\hat{\beta}(t)$ is calculated by averaging the last 100 draws, which adequately captures the true $\beta(t)$. Its 95% confidence interval (CI) covers the true function well, though it is much narrower at the beginning than it is at the end. This well known path degeneracy in the smoothing steps of SMC is still an open problem to be solved. The subject-specific predictions also agree well with their true values, which are not shown here.

The MLEs of the parameters and their SEs are displayed in Table 1 for both this model and the Poisson model. If of interest, the confidence intervals of the parameters can be constructed based on the asymptotic normal distributions. Though for the numeric examples in this paper, it is the

Table 1. MLEs and SEs

	Case 1		Case 2		Case 3	
Parameter	MLE	SE	MLE	SE	MLE	SE
$\log(\lambda_1)$	6.34	0.14	6.34	0.10	4.50	0.11
σ_1^2	2.03	0.52	2.57	0.63	0.12	0.028
σ_2^2	1.05×10^{-6}	0.27×10^{-6}	3.44×10^{-5}	0.59×10^{-5}	3.80×10^{-4}	0.72×10^{-4}
$\log(\lambda_2)$	6.21	0.026	7.44	0.016	3.69	0.020
σ_e^2	1.02	0.037	0.89	0.032		

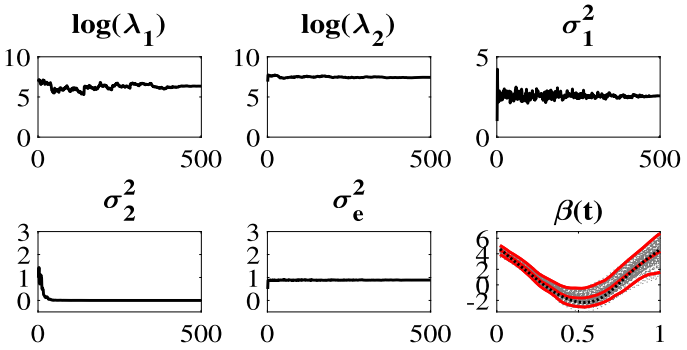

Figure 3. Case 2: Trace plots of parameters and the estimate of $\beta(t)$. For $\beta(t)$, the dotted black line is the true function, the solid red lines are the estimate and 95% CI, and the gray lines are the last 100 draws.

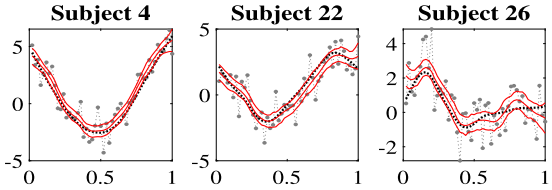

Figure 4. Case 2: Subject-specific fittings of three subjects. Gray dots connected by dotted lines are observations, black dotted lines are the true mean functions, and the red solid lines are estimated subject-specific means and 95% CIs.

function values not the parameter values that are of primary interest.

4.1.2 Case 2: SMC Gibbs SAEM

The number of inner iterations stayed the same as $n_q = 100$, while the outer iterations were reduced to $n_k = 500$ due to the long running-time particle algorithms. As shown in Table 2, each outer iteration took about 30 minutes for SMC Gibbs SAEM, while it only took less than 8 seconds for SAEM with exact Gibbs.

The trace plots in Figure 3 suggests that the algorithm has converged. For $\beta(t)$, the average values from the last 100 draws are shown as the red line, which is close to the true curve. The last 100 draw are also shown in the figure, which also show certain path degeneracy. Subject-specific fitting

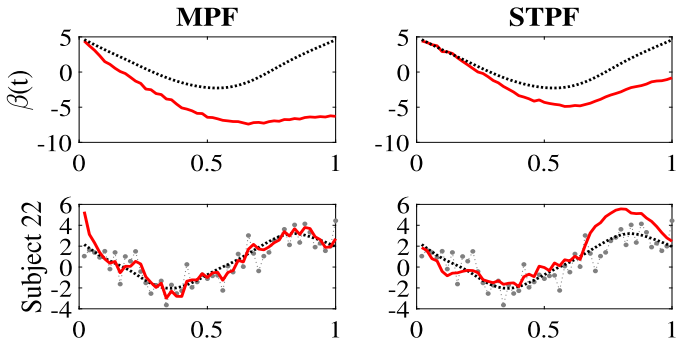

Figure 5. MPF and STPF for Gaussian FMM. The left column is for MPF; the right column is for STPF; the upper row is for $\beta(t)$; and the lower row is for Subject 22. The dotted black line is the true function, and the solid red line is the estimate.

is shown for three subjects in Figure 4. Though the overall trends are well captured, there are certain under-smoothing in some regions.

For MPF and STPF, the filtered means of $\beta(t)$ and Subject 22 are shown in Figure 5. Both MPF and STPF generate biased $\beta(t)$ estimates, which are more severe as t increases. The subject-specific fittings of Subject 22 show apparent undersmoothing.

4.2 Case 3: Poisson outcome

Let $\mu_{ij} = \mathrm{E}(y_{ij})$, a model for Poisson outcome with logarithm link function is used as follows

$$y_{ij}|\mu_{ij} \sim \text{Poisson}\left(\mu_{ij}\right), \tag{26}$$

$$\log\left\{\mu_{ij}|\beta\left(t_j\right), \alpha_i\left(t_j\right)\right\} = \beta\left(t_j\right) + \alpha_i\left(t_j\right), \tag{27}$$

with $n = 30$ and $T = 50$. To avoid extremely large Poisson outcome, we rescaled the curves in Figure 1 as shown in Figure 6. Similar to Case 2, we let $n_q = 100$ for inner iterations and $n_k = 500$ for outer iterations.

The results are displayed in Figure 7. The convergence situation shown by the trace plots is similar to Case 2. The estimate of $\beta(t)$ is obtained by averaging the last 100 draws, which shows some underestimate toward the end. A subject-specific fitting is displayed. Though the overall trend is captured, there is also some underestimate toward the end.

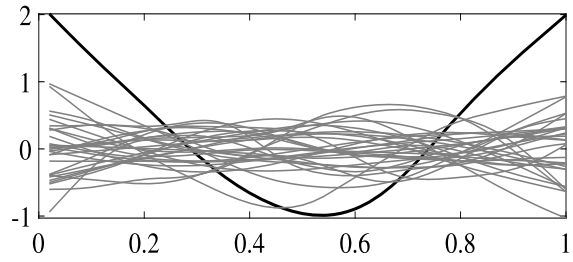

Figure 6. True functional effects for Poisson outcome: the thick black line is $\beta(t)$ and the thin gray lines are $\alpha_i(t)$'s.

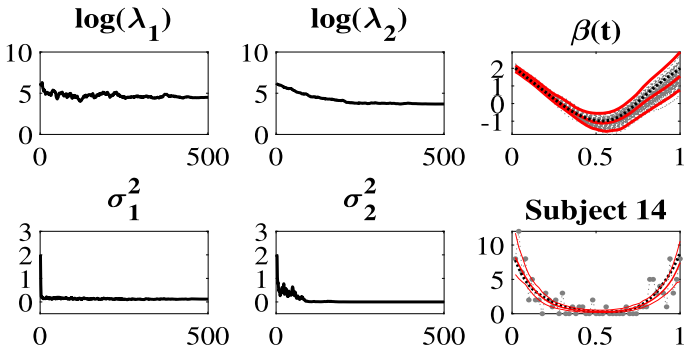

Figure 7. Case 3: Trace plots of parameter estimates, $\beta(t)$ and its estimate, and fitting for Subject 14. For $\beta(t)$, the dotted black line is the true function, the solid red line is the estimate, and the gray lines are the last 100 draws. For Subject 14, gray dots connected by dotted lines are observations, black dotted lines are the true mean functions, and the red solid lines are estimated subject-specific mean and CI.

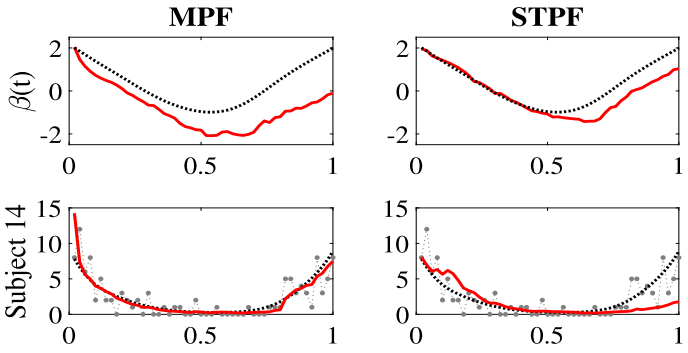

Figure 8. MPF and STPF for Poisson FMM. The left column is for MPF; the right column is for STPF; the upper row is for $\beta(t)$; and the lower row is for Subject 22. The dotted black line is the true function, and the solid red line is the estimate.

For MPF and STPF, the filtered means of $\beta(t)$ and Subject 22 are shown in Figure 8. Similar to Gaussian FMM results, both MPF and STPF generate biased $\beta(t)$ estimates, which are more severe as t increases. The subject-specific

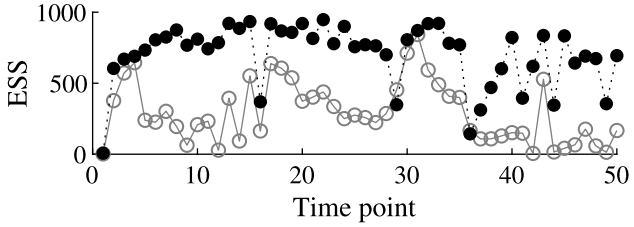

Figure 9. Effective sample sizes with $N = 1000$ particles: the circles with solid line are for the vector-wise approach, and the dots with dotted lines are for ϕ_j's from the proposed algorithm.

Table 2. Computational time (h as hours)

	Case 1	Case 2	Case 3
Outer iteration	5000	500	500
Inner iteration	100	100	100
CPU time	10.4h	258.6h	277.7h
Wall time	10.4h	88.3h	92.0h

fittings of Subject 14 also show undersmoothing and some bias.

4.3 ESS and computational time

For these examples, the dimensions of $\boldsymbol{\xi}_j$'s are $d = 62$, which is not large enough to cause severe particle collapse even if the SMC algorithms are directly applied to $\boldsymbol{\xi}_j$'s. Figure 9 shows that for the Poisson FMM example, the proposed algorithm has majority ESSs above 750 while the vector-wise approach has majority ESSs below 250. We further consider ESS < 10 an ad hoc definition of particle collapse for $N = 1000$ fixed number of particles. For $n = 30,\ 300,\ 1000,\ 3000$ subjects, there were 2, 12, 19, 25 particle collapses for the vector-wise approach respectively, and 0, 2, 3, 1 particle collapses for the proposed algorithm respectively, out of 50 time points.

Indiana University large-memory computer cluster, Carbonate, was used for computation. Each node of Carbonate is a Lenovo NeXtScale nx360 M5 server equipped with 256 GB of RAM, two 12-core Intel Xeon E5-2680 v3 CPUs and four 480 GB solid-state drives. With 2 nodes and 5 processors, the computational time for the three cases is summarized in Table 2.

5. A DATA EXAMPLE

In this section, we demonstrate the proposed algorithm in the nonlinear and non-Gaussian setting using an influenza example. The U.S. Outpatient Influenza-like Illness Surveillance Network (ILINet) data are downloaded from the website of Centers for Disease Control and Prevention at https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html. ILINet collects weekly state-level

outpatient influenza-like-illness (ILI) data, which is an important component of the U.S. Influenza Surveillance System. For this example, we focus on the ILI percentages from year 2011–2019 and 49 states after excluding Florida due to missing data. We further average every four-weeks to create pseudo months so that every year has 13 'month's. Overall, there are $n = 49$ subjects and $T = 117$ time points. In general, the flu season is considered as October through May, which does not cover the whole year. Instead, we use the calendar year as an ad hoc but simpler approach.

It is known that influenza infection rates exhibit an approximately seasonal pattern annually, while the overall levels may vary from year-to-year. To analyze the influenza data, we extend the method in [47] to the multiple-subject setting. [47] proposed that for a multiplicative model as $y_t = \omega_t \alpha_t + \varepsilon_t$, iterations between ω_t and α_t can be adopted for computations. For the influenza data, we propose an multiplicative model of the overall trend μ_t and subject-specific seasonal effect b_{it} as follows

$$y_{it} = \mu_t b_{it} + \varepsilon_{it}, \quad \varepsilon_{it} \sim \mathrm{N}\left(0, \sigma_e^2\right), \tag{28}$$

where ε_{it} is the error term. The multiplication of μ_t and b_{it} introduces both nonlinearity and non-Gaussianity. The trend μ_t shared by all the states is modeled by a random walk as $\mu_t = \mu_{t-1} + \varepsilon_{\mu t}$, where $\varepsilon_{\mu t} \sim \mathrm{N}\left(0, \sigma_\mu^2\right)$ for the first month of a year, and $\varepsilon_{\mu t} = 0$ for other months. We further force $\mu_t = 1$ for the first year. Consequently, μ_t is a step function with jumps at the beginning of each year, and its magnitudes represent the relative levels compared to the first year. The subject-specific seasonal effect b_{it} is modeled by a frequency model with $s = 13$ as $b_{it} = \sum_{j=1}^{6} \gamma_{ijt}$,

$$\begin{aligned}
\gamma_{ij,t+1} &= \gamma_{ijt}\cos\left(\lambda_j\right) + \gamma^*_{ijt}\sin\left(\lambda_j\right) + \omega_{ijt}, \\
\gamma^*_{ij,t+1} &= -\gamma_{ijt}\sin\left(\lambda_j\right) + \gamma^*_{ijt}\cos\left(\lambda_j\right) + \omega^*_{ijt},
\end{aligned}$$

$\lambda_j = 2\pi j/s$, for $j = 1, \cdots, 6$. The state vector $\boldsymbol{\alpha}_{it}$ is the collection of the six pairs of $\{\gamma_{ijt},\ \gamma^*_{ijt}\}$. The state transition matrix is a block diagonal matrix with six nonzero blocks as

$$\begin{pmatrix} \cos\lambda_j & \sin\lambda_j \\ -\sin\lambda_j & \cos\lambda_j \end{pmatrix}.$$

The state innovation $\boldsymbol{\omega}_{it}$ is the collection of the six pairs of $\{\omega_{ijt},\ \omega^*_{ijt}\}$ distributed as $\mathrm{N}(\mathbf{0}, \sigma_\omega^2 I_{12})$, which allows deviations from exact season patterns. The state is initialized at time zero as $\boldsymbol{\alpha}_{i0} \sim \mathrm{N}\left(\mathbf{0}, \sigma_0^2 I_{12}\right)$.

The proposed algorithm not only reduces the dimensions of the working state vectors from 589 to 1 for $\mu_t 4$ and 12 for b_{it}, but also the working state-space models become linear and Gaussian. Consequently, the exact draws in the inner iterations by [45, 46] are adopted instead of the conditional SMC updates. Figure 10 displays some results. The μ_t estimates suggest that compared year 2011, year 2012 had a slightly lower overall level and year 2016 had a much lower overall level, followed by higher levels for year 2017, 2018, and 2019. The state-level estimates adequately captured the individual states' rates, where the Indiana state is displayed as an example.

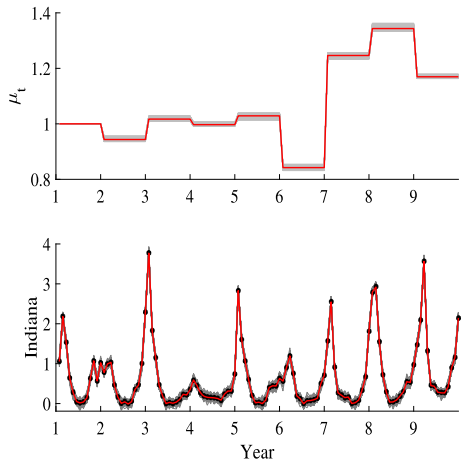

Figure 10. Influenza example: The upper panel displays 100 draws of μ_t (gray lines) and their averages (red line). The lower panel displays the Indiana state data (black dots), the 100 draws of the underlying mean values (gray lines) and their averages (red line).

6. DISCUSSION

The proposed algorithm provides a unified framework for state vector estimation, hence functional effects estimation, parameter estimation and inference. The numeric examples show that the algorithm produces reasonable results. However, the computational burden as shown in Table 2 is still intense, which imposes a challenge to application to real functional data with many subjects. It is straightforward to see that the computational cost for a single inner iteration is linear in the sample sizes. For example for the Poisson FMM for $n = 30,\ 300,\ 1000,\ 3000$ subjects, a single inner iteration took 6, 54, 179, and 536 seconds, respectively. [48] showed that for convergence, the number of Gibbs iterations needed is in the order of logarithms of location parameters in variance component models. If similar results apply to FMM, a computational cost of $O(n\log(n))$ would be needed for the proposed algorithm.

Potentially there are several ways to improve the computational efficiency. The numeric examples were programmed in Matlab (The Mathworks, Inc., Natick, MA, USA). Using other programming languages, such as C++, may improve the computational efficiency. Parallel computing, which is an active research area in SMC [49], may also be able to improve the computational efficiency. Also, we rely on [23, 25] for theoretical justification. A detailed theoretical analysis of the proposed algorithm may be helpful to improve the computational efficiency.

APPENDIX A. M-STEPS

The M-step for the linear Gaussian model is summarized in this appendix. The M-step for the Poisson model is similar, except that the Poisson model does not have σ_e^2. Consequently, one fewer item is needed for both updating $\boldsymbol{s}_k^*$ and $\boldsymbol{\theta}(k)$ for the Poisson model.

By ignoring the constant part with respect to $\boldsymbol{\theta}$, the logarithm of the complete data likelihood is as follows

$$\begin{aligned} l_c = & -T\log(\lambda_1) - nT\log(\lambda_2) \\ & - \frac{1}{2}\lambda_1^{-1}\sum_{j=1}^{T}(\boldsymbol{\phi}_j - H\boldsymbol{\phi}_{j-1})^\top \Sigma^{-1}(\boldsymbol{\phi}_j - H\boldsymbol{\phi}_{j-1}) \\ & - \frac{1}{2}n\log(\sigma_1^2) - \frac{1}{2}n\log(\sigma_2^2) \\ & - \frac{1}{2\sigma_1^2}\sum_{i=1}^{n}\boldsymbol{\psi}_{i0(1)}^2 - \frac{1}{2\sigma_2^2}\sum_{i=1}^{n}\boldsymbol{\psi}_{i0(2)}^2 \\ & - \frac{1}{2}\lambda_2^{-1}\sum_{i=1}^{n}\sum_{j=1}^{T}(\boldsymbol{\psi}_{ij} - H\boldsymbol{\psi}_{i,j-1})^\top \Sigma^{-1}(\boldsymbol{\psi}_j - H\boldsymbol{\psi}_{i,j-1}) \\ & - \frac{1}{2}nT\log(\sigma_e^2) - \frac{1}{2\sigma_e^2}\sum_{i=1}^{n}\sum_{j=1}^{T}\{y_{ij} - \beta(t_j) - \alpha_i(t_j)\}^2, \end{aligned}$$

where $\boldsymbol{\psi}_{i0(1)}$ and $\boldsymbol{\psi}_{i0(2)}$ are the first and second elements of $\boldsymbol{\psi}_{i0}$, respectively. Let $\boldsymbol{s} = (\ s_1 \quad s_2 \quad s_3 \quad s_4 \quad s_5\ )^\top$ with

$$\begin{aligned} s_1 &= \sum_{j=1}^{T}(\boldsymbol{\phi}_j - H\boldsymbol{\phi}_{j-1})^\top \Sigma^{-1}(\boldsymbol{\phi}_j - H\boldsymbol{\phi}_{j-1}) \\ s_2 &= \sum_{i=1}^{n}\boldsymbol{\psi}_{i0(1)}^2, \quad s_3 = \sum_{i=1}^{n}\boldsymbol{\psi}_{i0(2)}^2, \\ s_4 &= \sum_{i=1}^{n}\sum_{j=1}^{T}(\boldsymbol{\psi}_{ij} - H\boldsymbol{\psi}_{i,j-1})^\top \Sigma^{-1}(\boldsymbol{\psi}_{ij} - H\boldsymbol{\psi}_{i,j-1}), \\ s_5 &= \sum_{i=1}^{n}\sum_{j=1}^{T}\{y_{ij} - \beta(t_j) - \alpha_i(t_j)\}^2. \end{aligned}$$

Stochastic approximation of the Q function is essentially stochastically approximating $\boldsymbol{s}$. Let $\boldsymbol{s}_k$ be the $\boldsymbol{s}$ vector evaluated using the k step draws, we have

$$\hat{\boldsymbol{s}}_k^* = (1-\gamma_k)\,\hat{\boldsymbol{s}}_{k-1}^* + \gamma_k \boldsymbol{s}_k,$$

with $\boldsymbol{s}_k^* = (\ s_{k1}^* \quad s_{k2}^* \quad s_{k3}^* \quad s_{k4}^* \quad s_{k5}^*\ )^\top$. Take the first derivative of $\hat{Q}_k$ with respect to $\boldsymbol{\theta}$, set them to zero, and solve for the update of $\boldsymbol{\theta}$ leads to

$$\begin{aligned} \lambda_1(k) &= \frac{s_{k1}^*}{2T}, \quad \lambda_2(k) = \frac{s_{k4}^*}{2nT}, \\ \sigma_1^2(k) &= \frac{s_{k2}^*}{n}, \quad \sigma_2^2(k) = \frac{s_{k3}^*}{n}, \quad \sigma_e^2(k) = \frac{s_{k5}^*}{nT}. \end{aligned}$$

APPENDIX B. STANDARD ERRORS

We use the linear Gaussian model for illustration. The items for the Poisson model are similar by removing items related to σ_e^2. For numeric stability considerations, SEs of $\theta_1 = \log(\lambda_1)$ and $\theta_4 = \log(\lambda_2)$ are calculated instead of the original scale. Let $\boldsymbol{s} = (\ s_1 \quad s_2 \quad s_3 \quad s_4 \quad s_5\ )^\top$ as in the previous section. The logarithm of the complete data is

$$\begin{aligned} l_c = & -T\theta_1 - \frac{1}{2}\exp(-\theta_1)\,s_1 \\ & - \frac{1}{2}n\log(\sigma_1^2) - \frac{1}{2\sigma_1^2}s_2 - \frac{1}{2}n\log(\sigma_2^2) - \frac{1}{2\sigma_2^2}s_3 \\ & - nT\theta_4 - \frac{1}{2}\exp(-\theta_4)\,s_4 - \frac{1}{2}nT\log(\sigma_e^2) - \frac{1}{2\sigma_e^2}s_5. \end{aligned}$$

The first and second derivatives, which are needed in the stochatic approximation of the observed information matrix, are as follows

$$\begin{aligned} \frac{\partial l_c}{\partial\theta_1} &= -T + \frac{1}{2}\exp(-\theta_1)\,s_1, \\ \frac{\partial^2 l_c}{\partial\theta_1^2} &= -\frac{1}{2}\exp(-\theta_1)\,s_1, \\ \frac{\partial l_c}{\partial\sigma_1^2} &= -\frac{1}{2}n(\sigma_1^2)^{-1} + \frac{1}{2}s_2(\sigma_1^2)^{-2}, \\ \frac{\partial^2 l_c}{\partial(\sigma_1^2)^2} &= \frac{1}{2}n(\sigma_1^2)^{-2} - s_2(\sigma_1^2)^{-3}, \\ \frac{\partial l_c}{\partial\sigma_2^2} &= -\frac{1}{2}n(\sigma_2^2)^{-1} + \frac{1}{2}s_3(\sigma_2^2)^{-2}, \\ \frac{\partial^2 l_c}{\partial(\sigma_2^2)^2} &= \frac{1}{2}n(\sigma_2^2)^{-2} - s_3(\sigma_2^2)^{-3}, \\ \frac{\partial l_c}{\partial\theta_4} &= -nT + \frac{1}{2}\exp(-\theta_4)\,s_4, \\ \frac{\partial^2 l_c}{\partial\theta_4^2} &= -\frac{1}{2}\exp(-\theta_4)\,s_4, \\ \frac{\partial l_c}{\partial(\sigma_e^2)} &= -\frac{1}{2}nT(\sigma_e^2)^{-1} + \frac{1}{2}s_5(\sigma_e^2)^{-2}, \\ \frac{\partial^2 l_c}{\partial(\sigma_e^2)^2} &= \frac{1}{2}nT(\sigma_e^2)^{-2} - s_5(\sigma_e^2)^{-3}. \end{aligned}$$

ACKNOWLEDGEMENTS

We thank the Indiana University High Performance Computing Systems team for providing the computational infrastructure and technical support.

Received 24 February 2020

REFERENCES

[1] Guo, W. (2002). Functional mixed effects models. *Biometrics*, **58**, 121–128. MR1891050

[2] Kapur, K., Sanchez, B., Pacheck, A., Darras, B., Rutkove, S. B. and Selukar, R. (2019). Functional mixed-effects modeling of longitudinal Duchenne muscular dystrophy electrical impedance myography data using state-space approach. *IEEE Transactions on Biomedical Engineering*, **66**, 1761–1768.

[3] Liu, Z., Cappola, A. R., Crofford, L. J. and Guo, W. (2014). Modeling bivariate longitudinal hormone profiles by hierarchical state space models. *Journal of the American Statistical Association*, **109**, 108–118. MR3180550

[4] Liu, Z. and Guo, W. (2015). Modeling diurnal hormone profiles by hierarchical state space models. *Statistics in Medicine*, **34**, 3223–3234. MR3412628

[5] Zhu, B., Song, P. X. and Taylor, J. M. (2011). Stochastic functional data analysis: a diffusion model-based approach. *Biometrics*, **67**, 1295–1304. MR2872379

[6] Kliethermesa, S. and Olesonb, J. (2014). A Bayesian approach to functional mixed-effects modeling for longitudinal data with binomial outcomes. *Statistics in Medicine*, **33**, 3130–3146. MR3260533

[7] Liu, D., Lu, T., Niu, X. F. and Wu, H. (2011). Mixed-effects state space models for analysis of longitudinal dynamic systems. *Biometrics*, **67**, 476–485. MR2829016

[8] Liu, Z. (2015). A particle filter approach to multiprocess dynamic models with application to hormone data. *Statistics in Biosciences*, **7**, 379–393.

[9] Yang, M., Cavanaugh, J. E. and Zamba, G. (2015). State-space models for count time series with excess zeros. *Statistical Modelling*, **15**, 70–90. MR3306578

[10] Gordon, N. J., Salmond, D. J. and Smith, A. F. M. (1993). Novel approaches to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F Radar and Signal Processing*, **140**, 107–113.

[11] Doucet, A., de Freitas, N. and Gordon, N. (Ed.) (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York. MR1847784

[12] Kong, A., Liu, J. S. and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, **89**, 278–288.

[13] Liu, J. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, **93**, 1032–1044. MR1649198

[14] Anderson, J. L. and Anderson, S. L. (1999). A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, **127**, 2741–2758.

[15] Snyder, C., Bengtsson, T., Bickel, P. and Anderson, J. (2008). Obstacles in high-dimensional particle filtering. *Monthly Weather Review*, **136**, 4629–4640.

[16] Bengtsson, T., Bickel, P. and Li, B. (2008). Curse-of-dimensionality revisited: collapse of the particle filter in very large scale systems. *Probability and Statistics: Essays in Honor of David A. Freedman*, Nolan, D. and Speed, T. Ed., Vol. 2, Institute of Mathematical Statistics, 316–334. MR2459957

[17] Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D. and Stuart, A. M. (2017). Importance sampling: intrinsic dimension and computational cost. *Statistical Science*, **32**, 405–431. MR3696003

[18] Djuric, P. M., Lu, T., and Bugallo, M. F. (2007). Multiple particle filtering. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, **3**, 1181–1184. MR0642901

[19] Beaudeau, J. P., Bugallo, M. F. and Djuric, P. M. (2015). RSSI-based multi-target tracking by cooperative agents using fusion of cross-target information. *IEEE Transactions on Signal Processing*, **63**, 5033–5044. MR3391718

[20] Rebeschini, P. and van Handel, R. (2015). Can local particle filters beat the curse of dimensionality? *The Annals of Applied Probability*, **25**, 2809–2866. MR3375889

[21] Murphy, J. and Godsill, S. J. (2016). Blocked particle Gibbs schemes for high dimensional interacting systems. *IEEE Journal of Selected Topics in Signal Processing*, **10**, 328–342.

[22] Beskos, A., Crisan, D., Jasra, A., Kamatani, K. and Zhou, Y. (2017). A stable particle filter for a class of high-dimensional state-space models. *Advances in Applied Probability*, **49**, 24–48. MR3631214

[23] Delyon, B., Lavielle, M. and Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, **27**, 94–128. MR1701103

[24] Kuhn, E. and Lavielle, M. (2004). Coupling a stochastic approximation version of EM with an MCMC procedure. *ESAIM: Probability and Statistics*, **8**, 115–131. MR2085610

[25] Andrieu, C., Doucet, A. and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**, 269–342. MR2758115

[26] Louis, T. A. (1982). Finding observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society: Series B*, **44**, 226–233. MR0676213

[27] Wecker, W. E. and Ansley, C. F. (1983). The signal extraction approach to nonlinear regression and spline smoothing. *Journal of the American Statistical Association*, **78**, 81–89. MR0696851

[28] Sallas, W. M. and Harville, D. A. (1981). Best linear recursive estimation for mixed linear models. *Journal of the American Statistical Association*, **76**, 860–869. MR0650897

[29] Harrison, P. J. and Stevens, C. F. (1971). A Bayesian approach to short-term forecasting. *Operational Research Quarterly*, **22**, 341–362. MR0292225

[30] Harrison, P. J. and Stevens, C. F. (1976). Bayesian forecasting (with discussion). *Journal of the Royal Statistical Society: Series B*, **38**, 204–247. MR0655429

[31] Ansley, C. F., Kohn, R. and Wong, C. M. (1993). Nonparametric spline regression with prior information. *Biometrika*, **80**, 75–88. MR1225215

[32] Prado, R. and West, M. (2010). *Time series modeling, computation, and inference*. Boca Raton, FL: CRC Press. MR2655202

[33] Durbin, J. and Koopman, S. J. (2012). *Time series analysis by state space methods*. Oxford University Press, 2nd ed. MR3014996

[34] Malik, S. and Pitt, M. K. (2011). Particle filters for continuous likelihood evaluation and maximization. *Journal of Econometrics*, **165**, 190–209. MR2846644

[35] Creal, D. (2012). A survey of sequential Monte Carlo methods for economics and finance. *Econometric Reviews*, **31**, 245–296.

[36] Li, T., Bolic, M. and Djuric, P. M. (2015). Resampling methods for particle filtering. *IEEE Signal Processing Magazine*, **32**, 70–86.

[37] Fearnhead, P. and Kunsch, H. R. (2018). Particle filter and data assimilation. *Annual Review of Statistics and Its Applications*, **5**, 421–449. MR3774754

[38] Brooks, S. P. and Gelman, A. (1996). General Methods for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics*, **7**, 434–455. MR1665662

[39] Jank, W. (2006). Implementing and diagnosing the stochastic approximation EM algorithm. *Journal of Computational and Graphical Statistics*, **15**, 803–829. MR2297632

[40] Schön, T. B., Wills, A. and Ninness, B. (2011). System identification of nonlinear state-space models. *Automatica*, **47**, 39–49. MR2878244

[41] Delattre, M. and Lavielle, M. (2013). Coupling the SAEM algorithm and the extended Kalman filter for maximum likelihood estimation in mixed-effects diffusion models. *Statistics and Its Inference*, **6**, 519–532. MR3164656

[42] Picchini, U. and Samson, A. (2018). Coupling stochastic EM and approximate Bayesian computation for parameter inference in state-space models. *Computational Statistics*, **33**, 179–212. MR3754714

[43] Gu, C. (2002). *Smoothing Spline ANOVA Models*. Springer-Verlag, New York. MR1876599

[44] Godsill, S. J., Doucet, A. and West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, **99**, 156–168. MR2054295

[45] Carter, C. K. and Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika*, **81**, 541–553. MR1311096

[46] de Jone, P. and Shephard, N. (1995). The simulation smoother for time series models. *Biometrika*, **82**, 339–350. MR1354233

[47] Shephard, N. (1994). Partial non-Gaussian state space. *Biometrika*, **81**, 115–131. MR1279661
[48] Rosenthal, J. S. (1993). Rates of convergence for Gibbs sampling for variance component models. *Annals of Statistics*, **23**, 740–761. MR1345197
[49] Nicely, M. A. and Wells, B. E. (2019). Improved parallel resampling methods for particle filtering. *IEEE Access*, **7**, 47593–47604.

Ziyue Liu
Indiana University School of Medicine
Department of Biostatistics
410 W. 10th Street, Suite 3000
Indianapolis, IN 46202
USA
E-mail address: ziliu@iu.edu