Online Semiparametric Regression via Sequential Monte Carlo

MARIANNE MENICTAS[1], CHRIS J. OATES[2] & MATT P. WAND[3]

[1]*Grubhub Inc., U.S.A.,* [2]*Newcastle University, U.K.*
and [3]*University of Technology Sydney, Australia*

26th August, 2024

Abstract

We develop and describe online algorithms for performing online semiparametric regression analyses. Earlier work on this topic is in Luts, Broderick & Wand (*J. Comput. Graph. Statist.*, 2014) where online mean field variational Bayes was employed. In this article we instead develop sequential Monte Carlo approaches to circumvent well-known inaccuracies inherent in variational approaches. For Gaussian response semiparametric regression models our new algorithms share the online mean field variational Bayes property of only requiring updating and storage of sufficient statistics quantities of streaming data. In the non-Gaussian case accurate online semiparametric regression requires the full data to be kept in storage. The new algorithms allow for new options concerning accuracy/speed trade-offs for online semiparametric regression.

Keywords: Generalized additive models; generalized linear mixed models; real-time algorithms; penalized splines.

1 Introduction

Online semiparametric regression is concerned with rapid online fitting of flexible regression models, such as generalized additive models, and continuously updated inference as data stream in. Luts, Broderick & Wand (2014) laid out a framework for online, also known as real-time, semiparametric regression in the wake of developments over the preceding two decades such as Bayesian mixed model-based penalized splines and mean field variational Bayes. The setting and motivations are identical to those of Luts *et al.* (2014) and background material in the earlier article on the essence of online semiparametric regression also applies here. The crux of this article is provision of sequential Monte Carlo alternatives to the online mean field variational Bayes approach of Luts *et al.* (2014).

Online fitting of statistical models for sequentially arriving data has a very long history and large literature. The essential goal is that of obtaining fits and corresponding inference with online updating – such that the online results are similar to the batch results. Clearly online fitting is preferable in applications in which the data arrive rapidly, and repeated batch fitting is not computationally feasible. Three recent examples of such situations are online fitting of regression models for streaming data such as electronic health records and mobile health data (Luo & Song, 2023), online anomaly detection in streaming temporal data (Talagala *et al.*, 2020) and real-time fitting of item response theory models for ratings of movies (Weng & Coad, 2018).

For semiparametric regression and related areas there is also a large literature on online fitting, with early contributions such as recursive kernel density estimation (e.g. Yamato, 1971; Carroll, 1976) and kernel regression (e.g. Krzyzak & Pawlak, 1982; Yin & Yin, 1996). However, none of these 20th Century contributions addressed the problem of online smoothing parameter choice and, instead, were concerned with the theoretical properties of kernel estimators for deterministic smoothing parameter sequences. Bayesian computing developments since the 1990s have given rise to online semiparametric regression schemes for which the smoothing

parameters are updated in a principled and practical manner. For the kriging approach to nonparametric regression, Gramacy & Polson (2011) achieve this via sequential Monte Carlo. As mentioned earlier, Luts *et al.* (2014) achieved it via mean field variational Bayes. The essence of the present article is sequential Monte Carlo methodology for online semiparametric regression according to the mixed model-based splines approach advocated in the Ruppert, Wand & Carroll (2003) monograph. This approach has the attraction of only requiring the mixed model extension of ordinary linear models to achieve nonparametric and semiparametric regression fitting and inference.

The algorithms of Luts *et al.* (2014) have the attractiveness of being *purely* online in that, when a new vector of observations arrives, the approximate Bayesian semiparametric regression fit is updated *without having to store or access previous observations*. Instead, only key sufficient statistics need to be updated – after which the new observation vector can be discarded. A disadvantage of Luts *et al.* (2014) is that the inference is subject to varying degrees of inaccuracy due to mean field-type variational approximation error. This is particularly the case for regression models with non-Gaussian responses.

The sequential Monte Carlo-based approach used here is devoid of variational approximations and produces accurate online semiparametric regression fitting and inference. In the case of regression models with Gaussian response, purely online fitting and inference is achievable. However, for regression models with non-Gaussian responses the purely online feature has to be sacrificed to overcome the accuracy shortcomings of the Luts *et al.* (2014) approach and the full data must be kept in storage. The upshot is that this article's online semiparametric regression approach is more accurate, but not as fast, as the approach used in Luts *et al.* (2014). Depending on the speed requirements of the application and volume of data requiring storage, the new sequential Monte Carlo approaches to online semiparametric regression may be preferable. In short, the contributions of this article provide users with speed/accuracy trade-off options for online semiparametric regression.

Sequential Monte Carlo methodology of the type used here originates with Kong *et al.* (1994). Lie & Chen (1998) applied the approach to dynamical systems and introduced the *sequential Monte Carlo* idiom. Other key early contributions include Gilks & Berzuini (2001) and Pitt & Shephard (1999). A general theoretical framework for sequential Monte Carlo was devised by Del Moral *et al.* (2006). A comprehensive and contemporary overview of sequential Monte Carlo is provided by Chopin & Papaspiliopoulos (2020).

Section 2 lays down some preliminary infrastructure that is intrinsic to the sequential Monte Carlo approach to online semiparametric regression. In Section 3 we treat Gaussian response models, starting with multiple linear regression. For this special case the new methodology is relatively simple and the essence of the general approach can be elucidated in a reasonably concise manner. Section 4 then tackles the more challenging non-Gaussian response situation. In Section 5 we present some illustrations of that demonstrate good inferential accuracy of online sequential Monte Carlo and contrast it with the patchy performance of online mean field variational Bayes. Some concluding remarks are made in Section 6.

2 Preliminary Infrastructure

Online semiparametric regression via sequential Monte Carlo depends on some fundamental concepts and results, which we lay out in this section. Throughout this section $I(\mathcal{P})$ denotes the indicator of the proposition $\mathcal{P}$ being true.

2.1 Discrete Distribution Nomenclature

Suppose that a discrete random variable assumes the values of 5, 11 and 13 with probabilities $\frac{2}{7}$, $\frac{4}{7}$ and $\frac{1}{7}$ respectively. Its *probability mass function*, $\mathfrak{p}$, is:

$$\mathfrak{p}(5) = \tfrac{2}{7}, \quad \mathfrak{p}(11) = \tfrac{4}{7}, \quad \mathfrak{p}(13) = \tfrac{1}{7} \quad \text{and} \quad \mathfrak{p}(x) = 0 \text{ if } x \notin \{5, 11, 13\}. \tag{1}$$

We say that $\mathfrak{p}$ has *atoms* $\boldsymbol{a} = (5, 11, 13)$ and *probabilities* $\boldsymbol{p} = (\frac{2}{7}, \frac{4}{7}, \frac{1}{7})$. The corresponding *cumulative distribution function* is

$$F(x; \boldsymbol{a}, \boldsymbol{p}) = \tfrac{2}{7} I(x \leq 5) + \tfrac{4}{7} I(x \leq 11) + \tfrac{1}{7} I(x \leq 13),\ x \in \mathbb{R}$$

and *quantile function* is

$$Q(q; \boldsymbol{a}, \boldsymbol{p}) \equiv \inf\{x \in \mathbb{R} : q \leq F(x)\} = 5 + 6I(q > \tfrac{2}{7}) + 2I(q > \tfrac{6}{7}),\ 0 \leq q \leq 1. \tag{2}$$

The concepts illustrated here are, of course, very basic and straightforwardly extended to general discrete random variables.

2.2 Discrete Posterior Distribution Approximations

Let θ be a generic parameter in a Bayesian statistical model that takes values over a continuum such as $\mathbb{R}$, $\mathbb{R}_+$ or $[0, 1]$. Also, let $\boldsymbol{y}_{\text{curr}}$ denote the currently observed data. Then, within the Bayesian model, θ is a continuous random variable and its posterior distribution is characterized by the probability density function $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$. An intrinsic feature of online semiparametric regression via sequential Monte Carlo is sequential approximation of $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$ by *probability mass functions* as new observations arrive. To repeat: even though $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$ is a probability density function, it is sequentially approximated by probability mass functions as the data stream in.

Suppose that a new observation y_{new} has just been read in. The currently observed data is then updated according to $\boldsymbol{y}_{\text{curr}} \longleftarrow (\boldsymbol{y}_{\text{curr}}, y_{\text{new}})$. The current posterior density function of θ, $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$, is updated to be a probability mass function having atoms $\boldsymbol{a}_\theta$ and probabilities $\boldsymbol{p}_\theta$. Then the current posterior mean of θ is approximated by $(\boldsymbol{p}_\theta)^T \boldsymbol{a}_\theta$ and, for example, a current approximate 95% credible interval for θ is

$$\big(Q(0.025; \boldsymbol{a}_\theta, \boldsymbol{p}_\theta), Q(0.975; \boldsymbol{a}_\theta, \boldsymbol{p}_\theta)\big)$$

where $Q(\cdot; \boldsymbol{a}_\theta, \boldsymbol{p}_\theta)$ is the quantile function corresponding to the probability mass function having atoms $\boldsymbol{a}_\theta$ and probabilities $\boldsymbol{p}_\theta$.

2.3 The SystematicResample Algorithm

A fundamental component of sequential Monte Carlo procedures is that of drawing a sample from a d-variate discrete distribution having M atoms. The sample size is also M. Drawing a simple random sample is usually called *multinomial resampling* in the sequential Monte Carlo literature. However, in their Section 9.7, Chopin & Papaspiliopoulos (2020) advise against multinomial sampling due to its poor performance compared with other schemes. A simple alternative scheme is *systematic resampling*, which is the one that we adopt here. The operational steps are provided by the SystematicResample algorithm, listed as Algorithm 1, which involves storing the atoms as columns of a $d \times M$ matrix.

Algorithm 1 *The* SYSTEMATICRESAMPLE *algorithm.*

Inputs: $\boldsymbol{\Theta}$ $(d \times M)$, $\boldsymbol{p}$ $(M \times 1)$ such that all entries of $\boldsymbol{p}$ are non-negative and $\boldsymbol{p}^T\mathbf{1} = 1$

$\boldsymbol{\omega}_1 \longleftarrow$ the $M \times 1$ vector of cumulative sums of the entries of $\boldsymbol{p}$; $\boldsymbol{\omega}_2 \longleftarrow M\boldsymbol{\omega}_1$

$u \longleftarrow$ draw from the Uniform$(0, 1)$ distribution ; $\omega_3 \longleftarrow u$; $k \longleftarrow 1$

for $m = 1, \ldots M$

while $\{\omega_3 < (\boldsymbol{\omega}_2)_k\}$ $k \longleftarrow k + 1$; $(\boldsymbol{\iota})_m \longleftarrow k$; $\omega_3 \longleftarrow \omega_3 + 1$

$\boldsymbol{\Theta} \longleftarrow$ $d \times M$ matrix with the current columns of $\boldsymbol{\Theta}$ replaced by those indexed by $\boldsymbol{\iota}$

Output: $\boldsymbol{\Theta}$ $(d \times M)$

An example of the last step of SYSTEMATICRESAMPLE is as follows: if $d = 3$, $M = 5$ and $\boldsymbol{\iota} = (3, 3, 5, 2, 2)$ then the inputted matrix

$$\begin{bmatrix} 1 & 4 & 7 & 10 & 13 \\ 2 & 5 & 8 & 11 & 14 \\ 3 & 6 & 9 & 12 & 15 \end{bmatrix} \quad \text{is outputted as} \quad \begin{bmatrix} 7 & 7 & 13 & 4 & 4 \\ 8 & 8 & 14 & 5 & 5 \\ 9 & 9 & 15 & 6 & 6 \end{bmatrix}.$$

A justification of Algorithm 1 is given in Section S.2.1 of the supplement.

2.4 Distributional Definitions

The random variable x has an Inverse Gamma distribution with parameters κ and λ, written $x \sim$ Inverse-Gamma(λ, κ), if and only if its density function is

$$\mathfrak{p}(x) = \frac{\lambda^\kappa}{\Gamma(\kappa)}\, x^{-\kappa-1} \exp(-\lambda/x) I(x > 0).$$

In the semiparametric regression models to follow it is usual to place a Half Cauchy prior distribution on each of the standard deviation parameters. Let σ denote a typical standard deviation parameter. Then the notation $\sigma \sim$ Half-Cauchy(s) means that σ has density function

$$\mathfrak{p}(\sigma) = \frac{2I(\sigma > 0)}{\pi\, s\{1 + (\sigma/s)^2\}}.$$

However, via introduction of an auxiliary random variable a, we can express $\sigma \sim$ Half-Cauchy(s) as follows:

$$\sigma^2 | a \sim \text{Inverse-Gamma}(\tfrac{1}{2}, 1/a), \quad a \sim \text{Inverse-Gamma}(\tfrac{1}{2}, 1/s^2). \tag{3}$$

Representation (3) has the attraction of leading to draws from standard distributions in our sequential Monte Carlo schemes.

2.5 Vector Definitions and Conventions

The symbol $\mathbf{1}$ denotes a column vector having all entries equal to 1. If $\boldsymbol{a}$ is column vector then $\|\boldsymbol{a}\| = \sqrt{\boldsymbol{a}^T\boldsymbol{a}}$ denotes the Euclidean norm of $\boldsymbol{a}$. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are both $d \times 1$ vectors then $\boldsymbol{a} \odot \boldsymbol{b}$ denotes the $d \times 1$ vector containing the element-wise products of the entries of $\boldsymbol{a}$ and $\boldsymbol{b}$. Similarly $\boldsymbol{a}/\boldsymbol{b}$ is the $d \times 1$ vector of element-wise quotients. Also, if s is a scalar-to-scalar function then $s(\boldsymbol{a})$ is the $d \times 1$ vector containing the element-wise function evaluations. An example is $\exp\left([7 \;\; 4 \;\; 6]^T\right) = \left[\exp(7) \;\; \exp(4) \;\; \exp(6)\right]^T$. Also, $\max(\boldsymbol{a})$ denotes the largest entry in $\boldsymbol{a}$.

2.6 Overview of Online Semiparametric Regression via Sequential Monte Carlo

Loosely speaking, *semiparametric regression* involves extensions of parametric linear models in which non-linear effects are handled using suitable basis functions and penalization (e.g. Ruppert *et al.*, 2003). Special cases include nonparametric regression, generalized additive models, varying-coefficient models and generalized additive mixed models. The version of semiparametric regression which we use here is Bayesian models for which the non-linear effects correspond to mixed model-based penalized splines. As an illustrative example, consider the regression-type data set with responses $y_i \in \{0, 1\}$ and bivariate continuous predictors (x_{1i}, x_{2i}), $1 \leq i \leq n$. Then a *logistic additive model* is

$$\begin{aligned} &y_i|\boldsymbol{\beta}, \boldsymbol{u}_1, \boldsymbol{u}_2 \overset{\text{ind.}}{\sim} \text{Bernoulli}\left(\text{expit}\left(\beta_0 + \beta_1 x_{1i} + \sum_{k=1}^{K_1} u_{1k} z_{1k}(x_{1i}) + \beta_2 x_{2i} + \sum_{k=1}^{K_2} u_{2k} z_{2k}(x_{2i})\right)\right), \\ &\beta_0, \beta_1, \beta_2 \overset{\text{ind.}}{\sim} N(0, \sigma_\beta^2), \quad \boldsymbol{u}_r|\sigma_{ur}^2 \overset{\text{ind.}}{\sim} N(\boldsymbol{0}, \sigma_{ur}^2 \boldsymbol{I}_{K_r}), \quad \sigma_{ur}^2|a_{ur} \overset{\text{ind.}}{\sim} \text{Inverse-Gamma}(\tfrac{1}{2}, 1/a_{ur}), \\ &a_{ur} \overset{\text{ind.}}{\sim} \text{Inverse-Gamma}\big(\tfrac{1}{2}, 1/s_{\sigma^2}^2\big), \quad r = 1, 2, \end{aligned} \tag{4}$$

where $\overset{\text{ind.}}{\sim}$ stands for "independently distributed as" and $\text{expit}(x) \equiv 1/(1 + e^{-x})$. The functions $\{z_{1k}(\cdot) : 1 \leq k \leq K_1\}$ and $\{z_{2k}(\cdot) : 1 \leq k \leq K_2\}$ are suitable spline bases (see e.g. Wand & Ormerod, 2008) for the x_1 and x_2 non-linear effects. Also, $\sigma_\beta > 0$ and $s_{\sigma^2} > 0$ are user-specified hyperparameters. Figure 1 is a directed acyclic graph representation of (4) with, for example, $\boldsymbol{y}$ denoting the vector containing the y_i data. The $\boldsymbol{y}$ node is shaded to indicate that it corresponds to the observed data. Each of the other nodes require inference.

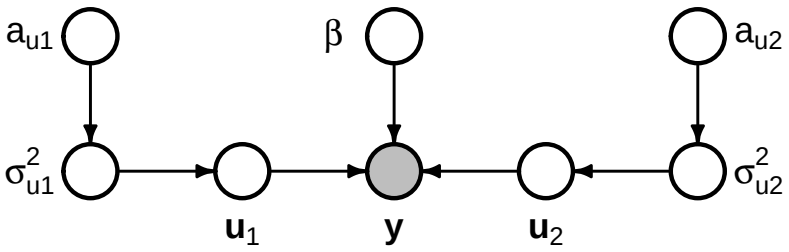

Figure 1: *Directed acyclic graph corresponding to the Bayesian logistic additive model (4). The shading indicates that the $\boldsymbol{y}$ node is observed.*

Model (4) and its Figure 1 representation exemplifies the approach to Bayesian semiparametric regression used here, with spline basis function penalization achieved via linear mixed model embedding. Batch fitting of (4) is straightforward using Bayesian inference engines such as JAGS (Plummer, 2022) and Stan (Stan Development Team, 2022) (e.g. Harezlak *et al.*, 2018). Our concern here, though, is online fitting of (4) as data stream in. Algorithm 5 of Luts *et al.* (2014) provides a solution to this problem using online mean field variational Bayes. However, this approach is susceptible to poor Bayesian inferential accuracy. Therefore, the sequential Monte Carlo alternative is being pursued here.

Typical Bayesian semiparametric regression models have between tens and hundreds of parameters requiring inference from the response and predictor observations. These include fixed effects, random effects, spline coefficients and covariance matrix parameters. Let d denote the total number of such variables and $\boldsymbol{\theta}$ be the $d \times 1$ vector containing them. For online semiparametric regression, the posterior density function of $\boldsymbol{\theta}$ is sequentially approximated by probability mass functions having M atoms, which are referred to as *particles*. The value of M is a user-specified tuning parameter and a reasonable default is $M = 1000$.

Figure 2 conveys the sequential Monte Carlo approach to online semiparametric regression in generic terms. Justification for this scheme, which applies to Bayesian models in general, is given in Section S.1 of the supplement. Most of the steps in Figure 2 involve simple calculations. The possible exception is the step involving drawing independent samples from the

Initialise the sample size to be 0. Initialise key sufficient statistic quantities.

Initialise the $d \times M$ matrix $\boldsymbol{\theta}_{\text{SMC}}$ such that each row contains M draws from the posterior distribution of each component of $\boldsymbol{\theta}$.

Initialise the particle probability vector $\boldsymbol{p}$ to be the $M \times 1$ vector with each entry equal to $1/M$.

Cycle:

Read in a new observation vector. Increment the sample size by 1.

Update key sufficient statistic quantities.

Use the likelihood of the new observation vector to update $\boldsymbol{p}$.

If the sum of squares of entries of $\boldsymbol{p}$ is above a particular threshold then

Update $\boldsymbol{\theta}_{\text{SMC}}$ by drawing a sample of size M from the d-variate discrete distribution with M atoms corresponding to the columns of $\boldsymbol{\theta}_{\text{SMC}}$ and probability vector $\boldsymbol{p}$. This step is facilitated by the SYSTEMATICRESAMPLE algorithm. Set $\boldsymbol{p}$ to be the $M \times 1$ vector with each entry equal to $1/M$.

Update $\boldsymbol{\theta}_{\text{SMC}}$ by drawing samples from the current full conditional distributions of sub-blocks of $\boldsymbol{\theta}$. Typically, the sub-blocks correspond to (1) the coefficients vector and (2) variance or covariance matrix parameters.

Approximate the current posterior distribution of $\boldsymbol{\theta}$ by the d-variate discrete distribution with M atoms corresponding to the columns of $\boldsymbol{\theta}_{\text{SMC}}$ and probability vector $\boldsymbol{p}$. Make inferential summaries of quantities of interest based on the current approximate posterior distribution of $\boldsymbol{\theta}$ as described in Section 2.2.

until data no longer available or analysis terminated.

Figure 2: *The sequential Monte Carlo approach to online semiparametric regression in generic form. Here $\boldsymbol{\theta}$ is the vector containing all fixed effects, random effects and covariance matrix parameters in the semiparametric regression model of interest.*

current full conditional distributions. For the sub-vectors of $\boldsymbol{\theta}$ for which the full conditional distribution has a standard form, such as Multivariate Normal or Inverse Gamma, this step is also straightforward. Moreover, for such fully Gibbsian settings, the updates only depend on sufficient statistics of the streaming data such as the sum of squares of the responses. Therefore, only these sufficient statistics need to be updated and stored for the Gibbsian situations that arise in Gaussian response semiparametric regression. The generalized response situation, with model (4) as an example, is more challenging due to the current full conditional distributions having non-standard forms and the need for more elaborate approaches that require passes through the current full data.

As explained in Section S.1.1.1 of the supplement, the general form of the probability vector updates when a new response y_{new} and its corresponding predictor data vector, arrives is

$$\boldsymbol{p}_m^{\text{new}} \propto \left\{ \frac{\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y}_{\text{curr}}, y_{\text{new}})}{\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y}_{\text{curr}})} \right\} \boldsymbol{p}_m^{\text{curr}}, \quad 1 \leq m \leq M. \tag{5}$$

Straightforward algebraic arguments then lead to the updating steps:

$$\begin{aligned} \boldsymbol{\ell}_m &\longleftarrow \boldsymbol{\ell}_m + \text{log-likelihood of } y_{\text{new}} \text{ based on } (\boldsymbol{\theta}_{\text{SMC}})_m, \quad 1 \le m \le M, \\ \boldsymbol{p}^{\text{new}} &\longleftarrow \frac{\exp\{\boldsymbol{\ell} - \max(\boldsymbol{\ell})\}}{\mathbf{1}^T \exp\{\boldsymbol{\ell} - \max(\boldsymbol{\ell})\}}, \end{aligned} \tag{6}$$

with logarithms and centring used to mitigate against overflow and underflow in the probability vector updates. Note that the likelihood of y_{new} is given by $\mathfrak{p}(y_{\text{new}}|\text{parents of } y_{\text{new}})$ in the model's directed acyclic graph. For the illustrative logistic additive model example given by (4) and Figure 1 we have

$$\boldsymbol{\ell}_m \longleftarrow \boldsymbol{\ell}_m + y_{\text{new}}\eta_m - \log(1 + e^{\eta_m}), \quad 1 \le m \le M,$$

where

$$\eta_m \equiv (\boldsymbol{\beta}_{0\text{SMC}})_m + (\boldsymbol{\beta}_{1\text{SMC}})_m x_{1\text{new}} + \sum_{k=1}^{K_1} (\boldsymbol{u}_{1\text{SMC}})_{km} z_{1k}(x_{1\text{new}}) + (\boldsymbol{\beta}_{2\text{SMC}})_m x_{2\text{new}} + \sum_{k=1}^{K_2} (\boldsymbol{u}_{2\text{SMC}})_{km} z_{2k}(x_{2\text{new}})$$

and $(x_{1\text{new}}, x_{2\text{new}})$ is the new predictor pair that partners y_{new}.

3 Gaussian Response Models

For reasons explained in Section 1, it is prudent to first describe the online semiparametric regression via sequential Monte Carlo for situations where Gaussianity of the responses can be assumed. We start with the familiar multiple linear regression setting.

3.1 Multiple Linear Regression

Let $\boldsymbol{X}$ be a $n \times p$ design matrix and consider the Bayesian regression model

$$\boldsymbol{y}|\,\boldsymbol{\beta}, \sigma^2 \sim N(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\,\boldsymbol{I}), \quad \boldsymbol{\beta} \sim N(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}), \quad \sigma \sim \text{Half-Cauchy}(s_{\sigma^2}) \tag{7}$$

As explained in Section 2.4 an equivalent, but more tractable model, is that where

$$\sigma \sim \text{Half-Cauchy}(s_{\sigma^2})$$

is replaced by the auxiliary variable representation

$$\sigma^2|\,a \sim \text{Inverse-Gamma}(\tfrac{1}{2}, 1/a), \quad a \sim \text{Inverse-Gamma}(\tfrac{1}{2}, 1/s_{\sigma^2}^2). \tag{8}$$

Batch fitting of (7) via Markov chain Monte Carlo is very established (e.g. Gelman *et al.*, 2014; Chapters 11–12) and it is listed in Algorithm 2. It relies on the result (e.g. Tierney, 1994) that, after convergence can be assumed following the "burn-in" phase of length N_{burn},

$$\begin{gathered} \text{successive draws from the full conditional distributions of } \boldsymbol{\beta}, \sigma^2 \text{ and } a \\ \text{constitute draws for the joint posterior density function: } \mathfrak{p}(\boldsymbol{\beta}, \sigma^2, a|\boldsymbol{y}). \end{gathered} \tag{9}$$

Note that Algorithm 2 uses the spectral decomposition of the matrix denoted by $\boldsymbol{\Omega}$ to efficiently obtain draws from the full conditional distribution of the $\boldsymbol{\beta}$ vector. The justifications for this and other aspects of Algorithm 2 are given in Section S.2.2 of the supplement.

Algorithm 3 is the online counterpart of Algorithm 2, based on the general approach of Figure 2. Algorithm 3 differs in that the data arrive sequentially and the fits and inferential summaries are updated in real time. It has the attractive feature that the posterior distribution updates depend only on the sufficient statistics **yTy**, **XTy** and **XTX**. This implies that the streaming data does not have to be stored or used again after the sufficient statistics have been updated. In this sense, Algorithm 3 achieves purely online fitting and inference according to

Algorithm 2 *Batch Markov chain Monte Carlo algorithm for approximate inference in the Gaussian response linear model.*

Data Inputs: $\boldsymbol{y}$ $(n \times 1)$ and $\boldsymbol{X}$ $(n \times p)$.

Markov Chain Monte Carlo Dimension Inputs: N_{burn} and N_{kept}, both positive integers.

Hyperparameter Inputs: $\boldsymbol{\mu_\beta}$ $(p \times 1)$, $\boldsymbol{\Sigma_\beta}$ $(p \times p)$ symmetric and positive definite, $s_{\sigma^2} > 0$.

$\mathbf{yTy} \longleftarrow \boldsymbol{y}^T\boldsymbol{y}$; $\mathbf{XTX} \longleftarrow \boldsymbol{X}^T\boldsymbol{X}$; $\mathbf{XTy} \longleftarrow \boldsymbol{X}^T\boldsymbol{y}$

For $g = 1, \ldots, N_{\text{burn}} + N_{\text{kept}}$:

$\boldsymbol{\Omega} \longleftarrow \dfrac{\mathbf{XTX}}{(\sigma^2)^{[g-1]}} + \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}$; decompose $\boldsymbol{\Omega} = \boldsymbol{U}_{\boldsymbol{\Omega}}\text{diag}(\boldsymbol{d}_{\boldsymbol{\Omega}})\boldsymbol{U}_{\boldsymbol{\Omega}}^T$ where $\boldsymbol{U}_{\boldsymbol{\Omega}}\boldsymbol{U}_{\boldsymbol{\Omega}}^T = \boldsymbol{I}$

$\boldsymbol{z} \longleftarrow p \times 1$ vector containing totally independent $N(0,1)$ draws

$$\boldsymbol{\beta}^{[g]} \longleftarrow \boldsymbol{U}_{\boldsymbol{\Omega}}\left[\frac{\boldsymbol{U}_{\boldsymbol{\Omega}}^T\boldsymbol{z}}{\sqrt{\boldsymbol{d}_{\boldsymbol{\Omega}}}} + \frac{\boldsymbol{U}_{\boldsymbol{\Omega}}^T\left\{\mathbf{XTy}/(\sigma^2)^{[g-1]} + \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}}\right\}}{\boldsymbol{d}_{\boldsymbol{\Omega}}}\right]$$

$$a^{[g]} \overset{\text{ind.}}{\sim} \text{Inverse-Gamma}\left(1, \{(\sigma^2)^{[g-1]}\}^{-1} + s_{\sigma^2}^{-1}\right)$$

$$(\sigma^2)^{[g]} \overset{\text{ind.}}{\sim} \text{Inverse-Gamma}\Big(\tfrac{1}{2}(n+1), (a^{[g]})^{-1} + \tfrac{1}{2}\left\{\mathbf{yTy} - 2(\mathbf{XTy})^T\boldsymbol{\beta}^{[g]} + \left(\boldsymbol{\beta}^{[g]}\right)^T\mathbf{XTX}\boldsymbol{\beta}^{[g]}\right\}\Big).$$

Produce summaries based on the kept $\boldsymbol{\beta}^{[g]}$ and $(\sigma^2)^{[g]}$ chains,
$N_{\text{burn}} + 1 \leq g \leq (N_{\text{burn}} + N_{\text{kept}})$, being draws from the posterior distributions of $\boldsymbol{\beta}$ and σ^2 (due to result (9)).

Algorithm 3 *Online sequential Monte Carlo algorithm for online approximate inference in the Gaussian response linear model.*

Tuning Parameter Inputs (defaults): $M \in \mathbb{N}$ (1000) ; $\tau > 0$ $(2/M)$.

Hyperparameter Inputs: $\boldsymbol{\mu}_{\boldsymbol{\beta}}$ $(p \times 1)$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ $(p \times p)$ symmetric and positive definite, $s_{\sigma^2} > 0$.

Initialize:

$\boldsymbol{\beta}_{\text{SMC}} \longleftarrow p \times M$ matrix with columns containing independent random samples from $N(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$

$\boldsymbol{a}_{\text{SMC}} \longleftarrow 1 \times M$ vector containing a random sample from Inverse-Gamma$(\frac{1}{2}, 1/s_{\varepsilon}^2)$

$\boldsymbol{\sigma}^2_{\text{SMC}} \longleftarrow 1 \times M$ vector containing a random sample from Half-Cauchy(s_{σ^2})

$\boldsymbol{\ell} \longleftarrow \log\left(\frac{1}{M}\right)\mathbf{1}$; $n \longleftarrow 0$; $\mathbf{yTy} \longleftarrow 0$; $\mathbf{XTy} \longleftarrow \mathbf{0}$ $(p \times 1)$; $\mathbf{XTX} \longleftarrow \mathbf{0}$ $(p \times p)$.

Cycle:

Read in $y_{\text{new}}(1 \times 1)$ and $\boldsymbol{x}_{\text{new}}(p \times 1)$; $n \longleftarrow n + 1$

$\mathbf{yTy} \longleftarrow \mathbf{yTy} + y^2_{\text{new}}$; $\mathbf{XTy} \longleftarrow \mathbf{XTy} + \boldsymbol{x}_{\text{new}} y_{\text{new}}$; $\mathbf{XTX} \longleftarrow \mathbf{XTX} + \boldsymbol{x}_{\text{new}} \boldsymbol{x}^T_{\text{new}}$

$\boldsymbol{\eta} \longleftarrow \boldsymbol{\beta}^T_{\text{SMC}} \boldsymbol{x}_{\text{new}}$; $\boldsymbol{\ell} \longleftarrow \boldsymbol{\ell} + (y_{\text{new}} \boldsymbol{\eta} - \frac{1}{2}\,\boldsymbol{\eta} \odot \boldsymbol{\eta})/\{(\boldsymbol{\sigma}^2_{\text{SMC}})^T\} - \frac{1}{2}\log\{(\boldsymbol{\sigma}^2_{\text{SMC}})^T\}$

$\boldsymbol{p} \longleftarrow \exp\{\boldsymbol{\ell} - \max(\boldsymbol{\ell})\}/[\mathbf{1}^T \exp\{\boldsymbol{\ell} - \max(\boldsymbol{\ell})\}]$

If $\boldsymbol{p}^T \boldsymbol{p} > \tau$ then

$$\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{a}_{\text{SMC}} \\ \boldsymbol{\sigma}^2_{\text{SMC}} \end{bmatrix} \longleftarrow \text{SystematicResample}\left(\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{a}_{\text{SMC}} \\ \boldsymbol{\sigma}^2_{\text{SMC}} \end{bmatrix}, \boldsymbol{p} \right) \quad ; \quad \boldsymbol{\ell} \longleftarrow \log\left(\frac{1}{M}\right)\mathbf{1}$$

For $m = 1, \ldots, M$:

$\boldsymbol{\Omega} \longleftarrow \dfrac{\mathbf{XTX}}{(\boldsymbol{\sigma}^2_{\text{SMC}})_m} + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{\beta}}$; decompose $\boldsymbol{\Omega} = \boldsymbol{U}_{\boldsymbol{\Omega}} \text{diag}(\boldsymbol{d}_{\boldsymbol{\Omega}}) \boldsymbol{U}^T_{\boldsymbol{\Omega}}$ where $\boldsymbol{U}_{\boldsymbol{\Omega}} \boldsymbol{U}^T_{\boldsymbol{\Omega}} = \boldsymbol{I}$

$\boldsymbol{z} \longleftarrow p \times 1$ vector containing totally independent $N(0,1)$ draws

$$m\text{th column of } \boldsymbol{\beta}_{\text{SMC}} \longleftarrow \boldsymbol{U}_{\boldsymbol{\Omega}} \left\{ \frac{\boldsymbol{U}^T_{\boldsymbol{\Omega}} \boldsymbol{z}}{\sqrt{\boldsymbol{d}_{\boldsymbol{\Omega}}}} + \frac{\boldsymbol{U}^T_{\boldsymbol{\Omega}} \Big(\mathbf{XTy} / (\boldsymbol{\sigma}^2_{\text{SMC}})_m + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{\beta}} \boldsymbol{\mu}_{\boldsymbol{\beta}} \Big)}{\boldsymbol{d}_{\boldsymbol{\Omega}}} \right\}$$

$$(\boldsymbol{a}_{\text{SMC}})_m \sim \text{Inverse-Gamma}\Big(1, (\boldsymbol{\sigma}^2_{\text{SMC}})^{-1}_m + s^{-1}_{\sigma^2}\Big)$$

$$(\boldsymbol{\sigma}^2_{\text{SMC}})_m \sim \text{Inverse-Gamma}\Big(\tfrac{1}{2}(n+1), (\boldsymbol{a}_{\text{SMC}})^{-1}_m + \tfrac{1}{2}\Big\{ \mathbf{yTy} - 2\big((\mathbf{XTy})^T \boldsymbol{\beta}_{\text{SMC}}\big)_m + \big(\boldsymbol{\beta}^T_{\text{SMC}} \mathbf{XTX} \boldsymbol{\beta}_{\text{SMC}}\big)_{mm} \Big\}\Big).$$

Produce summaries based on the current approximate posterior distributions of $\boldsymbol{\beta}$ and σ^2 equalling the probability mass functions with atoms stored in $\boldsymbol{\beta}_{\text{SMC}}$ and $\boldsymbol{\sigma}^2_{\text{SMC}}$, respectively, and probabilities $\boldsymbol{p}$.

until data no longer available or analysis terminated.

the definition described in Section 1. Section S.2.3 of the supplement provides justifications for the Algorithm 3 steps.

Algorithm 4 is a modification of Algorithm 3 that allows for the possibility of batched-based tuning at the start of the online regression analysis. Its justification is given in Section S.2.4 of the supplement. An illustration of batch-based tuning to properly initialise an online semiparametric regression analysis is given in Section 5.2 (see Figure 5).

Algorithm 4 *Modification of Algorithm 3 to include batch-based tuning and convergence diagnosis.*

1. Set n_{warm} to be the warm-up sample size and n_{valid} to be size of the validation period. Read in the first $n_{\text{warm}} + n_{\text{valid}}$ response and predictor values.
2. Create $\boldsymbol{y}_{\text{warm}}$ and $\boldsymbol{X}_{\text{warm}}$ consisting of the first n_{warm} response and predictor values.
3. Feed $\boldsymbol{y}_{\text{warm}}$ and $\boldsymbol{X}_{\text{warm}}$ into the batch Markov chain Monte Carlo Algorithm 2 with $N_{\text{kept}} = M$. Use the kept chains $\boldsymbol{\beta}^{[g]}$, $(\sigma^2)^{[g]}$ and $a^{[g]}$, $1 \leq g \leq M$, to initialise $\boldsymbol{\beta}_{\text{SMC}}$, $\boldsymbol{a}_{\text{SMC}}$ and $\boldsymbol{\sigma}^2_{\text{SMC}}$.
4. Set $\mathbf{yTy} \leftarrow \boldsymbol{y}^T_{\text{warm}}\boldsymbol{y}_{\text{warm}}$, $\mathbf{XTy} \leftarrow \boldsymbol{X}^T_{\text{warm}}\boldsymbol{y}_{\text{warm}}$, $\mathbf{XTX} \leftarrow \boldsymbol{X}^T_{\text{warm}}\boldsymbol{X}_{\text{warm}}$ and $n \leftarrow n_{\text{warm}}$.
5. Run the online sequential Monte Carlo Algorithm 3 until $n = n_{\text{warm}} + n_{\text{valid}}$.
6. Use convergence diagnostic graphics to assess whether the online parameters are converging to the batch parameters.
   (a) If not converging then return to Step 1 and increase n_{warm}.
   (b) If converging then continue running the online sequential Monte Carlo Algorithm 3 until data no longer available or analysis terminated.

Algorithm 3 can be used to produce convergence diagnostic graphics analogous to Figures 3 and 5 in Luts, Broderick & Wand (2014).

3.2 Linear Mixed Models

Our mixed model-based splines approach to Bayesian semiparametric regression makes use of the following class of linear mixed models:

$$\begin{aligned} \boldsymbol{y}|\,\boldsymbol{\beta}, \boldsymbol{u}, \sigma^2_\varepsilon &\sim N(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u}, \sigma^2_\varepsilon \boldsymbol{I}), \\ \boldsymbol{u}|\,\sigma^2_{u1}, \ldots, \sigma^2_{uR} &\sim N(\boldsymbol{0}, \text{blockdiag}(\sigma^2_{u1}\boldsymbol{I}_{K_1}, \ldots, \sigma^2_{uR}\boldsymbol{I}_{K_R})). \end{aligned} \tag{10}$$

Here $\boldsymbol{y}$ is an $n \times 1$ vector of response variables, $\boldsymbol{\beta}$ is a $p \times 1$ vector of fixed effects, $\boldsymbol{u}$ is a vector of random effects, $\boldsymbol{X}$ and $\boldsymbol{Z}$ are design matrices, σ^2_ε is the error variance and $\sigma^2_{u1}, \ldots, \sigma^2_{ur}$ are variance parameters corresponding to sub-blocks of $\boldsymbol{u}$ of size $K_1, \ldots, K_R$. We set the priors to be

$$\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}), \quad \sigma_{ur} \sim \text{Half-Cauchy}(s_{ur}),\ 1 \leq r \leq R, \quad \sigma_\varepsilon \sim \text{Half-Cauchy}(s_\varepsilon) \tag{11}$$

with the hyperparameters $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ symmetric and positive definite and $s_\varepsilon, s_{ur} > 0$ for $1 \leq r \leq R$. As in Section 3, we introduce the auxiliary variables

$$a_{ur} \sim \text{Inverse-Gamma}(\tfrac{1}{2}, 1/s^2_{ur}) \quad \text{and} \quad a_\varepsilon \sim \text{Inverse-Gamma}(\tfrac{1}{2}, 1/s^2_\varepsilon) \tag{12}$$

and use of the analogue of (8) to induce Half-Cauchy priors on the standard deviation parameters.

As described in Section 2 of Zhao, Staudenmayer, Coull & Wand (2006), model (10)–(11) covers several important special cases, including (with example number from Zhao *et al.* 2006 added):

- simple random effects models (Examples 1 and 2),
- cross random effects models (Example 3),
- nested random effects models (Example 4),
- generalized additive models (Example 6),
- semiparametric mixed models (Example 7),
- bivariate smoothing and geoadditive models extensions (Example 8).

Examples 2 and 6 of Zhao *et al.* (2006) involve 2×2 and 3×3 unstructured covariance matrix parameters which, strictly speaking, are not special cases of (11). However, as discussed in Section 3.3, the unstructured covariance matrix extension is quite straightforward.

Let

$$\boldsymbol{C} = [\boldsymbol{X}\ \boldsymbol{Z}]$$

be the combined design matrix in (10) and P be the number of columns in $\boldsymbol{C}$. Then each pass of the corresponding online sequential Monte Carlo algorithm involves arrival and processing of a new scalar response measurement, y_{new}, and a $P \times 1$ vector $\boldsymbol{c}_{\text{new}}$, corresponding to the new row of $\boldsymbol{C}$. This results in Algorithm 5 for purely online fitting of (10). Its justification is given in Section S.2.5 of the supplement.

3.3 Extension to Unstructured Covariance Matrices for Random Effects

A simple special case of (10) is the *random intercept model*, for which the first two hierarchical levels are set to

$$\begin{gathered} y_{ij}|\beta_0, \beta_1, U_i, \sigma_\varepsilon^2 \overset{\text{ind.}}{\sim} N(\beta_0 + U_i + \beta_1\, x_{ij}, \sigma_\varepsilon^2), \quad 1 \le i \le m, \quad 1 \le j \le n_i, \\ \text{and} \quad U_i\big|\sigma_u^2 \overset{\text{ind.}}{\sim} N(0, \sigma_u^2). \end{gathered} \tag{13}$$

The *random intercepts and slopes* extension of (13) is

$$\begin{gathered} y_{ij}|\beta_0, \beta_1, U_i, V_i, \sigma_\varepsilon^2 \overset{\text{ind.}}{\sim} N(\beta_0 + U_i + (\beta_1 + V_i)\, x_{ij}, \sigma_\varepsilon^2), \quad 1 \le i \le m, \quad 1 \le j \le n_i, \\ \text{and} \quad \begin{bmatrix} U_i \\ V_i \end{bmatrix} \Big| \boldsymbol{\Sigma} \overset{\text{ind.}}{\sim} N(\boldsymbol{0}, \boldsymbol{\Sigma}), \quad \text{where} \quad \boldsymbol{\Sigma} \equiv \begin{bmatrix} \sigma_u^2 & \rho_{uv}\, \sigma_u\, \sigma_v \\ \rho_{uv}\, \sigma_u\, \sigma_v & \sigma_v^2 \end{bmatrix} \end{gathered}$$

is an unstructured 2×2 covariance matrix. The conjugate prior for $\boldsymbol{\Sigma}$ is the Inverse Wishart distribution. Note that

$$\begin{gathered} \boldsymbol{\Sigma}|\, a_{uv1}, a_{uv2} \sim \text{Inverse-Wishart}\left(\nu + 1, 2\nu \begin{bmatrix} 1/a_{uv1} & 0 \\ 0 & 1/a_{uv2} \end{bmatrix}\right), \\ a_{uv1}, a_{uv2} \overset{\text{ind.}}{\sim} \text{Inverse-Gamma}(\tfrac{1}{2}, 1/s_{uv}), \quad \nu, s_{uv} > 0 \end{gathered}$$

provides a covariance matrix extension of $\sigma_u \sim \text{Half-Cauchy}(A_u)$. The choice $\nu = 2$ imposes a Uniform$(-1, 1)$ distribution on ρ_{uv} and Half-t_2 distributions on σ_u and σ_v. This is described in Huang & Wand (2013), including the definition of the Inverse-Wishart$(a, \boldsymbol{B})$ distribution. Extensions to models with larger unstructured covariance matrices is similar.

4 Generalized Response Models

We now switch attention to semiparametric regression models for which the response is non-Gaussian, which we will refer to as *generalized* response models, and include binary and count response types. We begin with generalized linear models, for which the gist of the generalized response extension can be conveyed without a big notational burden.

Algorithm 5 *Online sequential Monte Carlo algorithm for approximate inference in the Gaussian response linear mixed model (10).*

Tuning Parameter Inputs (defaults): $M \in \mathbb{N}\ (1000)$; $\tau > 0\ (2/M)$.

Hyperparameter Inputs: $\boldsymbol{\mu}_{\boldsymbol{\beta}}\ (p \times 1)$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}\ (p \times p)$ symmetric and positive definite, $s_{\sigma^2} > 0$, $s_{ur} > 0,\ 1 \leq r \leq R$.

Perform batch-based tuning runs analogous to those described in Algorithm 4 and determine a warm-up sample size n_{warm} for which convergence is validated.

Set $\boldsymbol{y}_{\text{warm}}$ and $\boldsymbol{C}_{\text{warm}}$ to be the response vector and design matrix based on the first n_{warm} observations. Then set $\mathbf{yTy} \leftarrow \boldsymbol{y}_{\text{warm}}^T \boldsymbol{y}_{\text{warm}}$, $\mathbf{CTy} \leftarrow \boldsymbol{C}_{\text{warm}}^T \boldsymbol{y}_{\text{warm}}$, $\mathbf{CTC} \leftarrow \boldsymbol{C}_{\text{warm}}^T \boldsymbol{C}_{\text{warm}}$, $n \leftarrow n_{\text{warm}}$.

Set the following matrices (with dimensions)

$$\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}\ (P \times M)\ ;\ \boldsymbol{\sigma}^2_{\varepsilon\text{SMC}}\ (1 \times M)\ ;\ a_{\varepsilon\text{SMC}}\ (1 \times M)\ ;\ \boldsymbol{\sigma}^2_{u\text{SMC}}\ (r \times M)\ ;\ \boldsymbol{a}_{u\text{SMC}}\ (r \times M)$$

such that each column is a random sample from the relevant approximate posterior distribution according to the batch Markov chain Monte Carlo samples based on the first n_{warm} observations.

Cycle:

Read in $y_{\text{new}}\ (1 \times 1)$ and $\boldsymbol{c}_{\text{new}}\ (P \times 1)$; $n \leftarrow n + 1$

$$\boldsymbol{\eta} \longleftarrow \begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}^T \boldsymbol{c}_{\text{new}}\ \ ;\ \boldsymbol{\ell} \longleftarrow \boldsymbol{\ell} + (y_{\text{new}}\boldsymbol{\eta} - \tfrac{1}{2}\,\boldsymbol{\eta} \odot \boldsymbol{\eta})/\{(\boldsymbol{\sigma}^2_{\varepsilon\text{SMC}})^T\} - \tfrac{1}{2}\log\{(\boldsymbol{\sigma}^2_{\varepsilon\text{SMC}})^T\}$$

$$\boldsymbol{p} \longleftarrow \exp\{\boldsymbol{\ell} - \max(\boldsymbol{\ell})\}/[\mathbf{1}^T \exp\{\boldsymbol{\ell} - \max(\boldsymbol{\ell})\}]$$

If $\boldsymbol{p}^T\boldsymbol{p} > \tau$ then

$$\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \\ \boldsymbol{\sigma}^2_{\varepsilon\text{SMC}} \\ a_{\varepsilon\text{SMC}} \\ \boldsymbol{\sigma}^2_{u\text{SMC}} \\ \boldsymbol{a}_{u\text{SMC}} \end{bmatrix} \longleftarrow \text{SYSTEMATICRESAMPLE}\left(\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \\ \boldsymbol{\sigma}^2_{\varepsilon\text{SMC}} \\ a_{\varepsilon\text{SMC}} \\ \boldsymbol{\sigma}^2_{u\text{SMC}} \\ \boldsymbol{a}_{u\text{SMC}} \end{bmatrix}, \boldsymbol{p} \right)\ ;\ \boldsymbol{\ell} \longleftarrow \log\left(\tfrac{1}{M}\right)\mathbf{1}$$

$\mathbf{yTy} \leftarrow \mathbf{yTy} + y^2_{\text{new}}$; $\mathbf{CTy} \leftarrow \mathbf{CTy} + \boldsymbol{c}_{\text{new}}\, y_{\text{new}}$; $\mathbf{CTC} \leftarrow \mathbf{CTC} + \boldsymbol{c}_{\text{new}}\, \boldsymbol{c}^T_{\text{new}}$

For $m = 1, \ldots, M$:

$$\boldsymbol{\Omega} \longleftarrow \frac{\mathbf{CTC}}{(\boldsymbol{\sigma}^2_{\varepsilon\text{SMC}})_m} + \text{blockdiag}\Big(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}, \boldsymbol{I}_{K_1}/(\boldsymbol{\sigma}^2_{u\text{SMC}})_{1m}, \ldots, \boldsymbol{I}_{K_R}/(\boldsymbol{\sigma}^2_{u\text{SMC}})_{Rm}\Big)$$

decompose $\boldsymbol{\Omega} = \boldsymbol{U}_{\boldsymbol{\Omega}}\text{diag}(\boldsymbol{d}_{\boldsymbol{\Omega}})\boldsymbol{U}_{\boldsymbol{\Omega}}^T$ where $\boldsymbol{U}_{\boldsymbol{\Omega}}\boldsymbol{U}_{\boldsymbol{\Omega}}^T = \boldsymbol{I}$

$\boldsymbol{z} \longleftarrow P \times 1$ vector containing totally independent $N(0,1)$ draws

$$m\text{th column of} \begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix} \longleftarrow \boldsymbol{U}_{\boldsymbol{\Omega}} \left[\frac{\boldsymbol{U}_{\boldsymbol{\Omega}}^T \boldsymbol{z}}{\sqrt{\boldsymbol{d}_{\boldsymbol{\Omega}}}} + \frac{\boldsymbol{U}_{\boldsymbol{\Omega}}^T \Big\{ \mathbf{CTy}/(\boldsymbol{\sigma}^2_{\varepsilon\text{SMC}})_m + \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} \Big\}}{\boldsymbol{d}_{\boldsymbol{\Omega}}} \right]$$

continued on a subsequent page ...

Algorithm 5 continued. *This is a continuation of the description of this algorithm that continues on a preceding page.*

$(a_{\varepsilon\text{SMC}})_m \sim \text{Inverse-Gamma}\left(1, (\boldsymbol{\sigma}^2_{\varepsilon\text{SMC}})_m^{-1} + s_\varepsilon^{-1}\right)$

$(\boldsymbol{\sigma}^2_{\varepsilon\text{SMC}})_m \sim \text{Inverse-Gamma}\Bigg(\frac{1}{2}(n+1), (a_{\varepsilon\text{SMC}})_m^{-1}$

$+\frac{1}{2}\left\{\mathbf{yTy} - 2\left((\mathbf{CTy})^T \begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}\right)_m + \left(\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}^T \mathbf{CTC} \begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}\right)_{mm}\right\}\Bigg)$

$i_{\text{stt}} \longleftarrow 1$

For $r = 1, \ldots, R$:

$(\boldsymbol{a}_{u\text{SMC}})_{rm} \sim \text{Inverse-Gamma}\left(1, (\boldsymbol{\sigma}^2_{u\text{SMC}})_{rm}^{-1} + s_{ur}^{-1}\right)$

$i_{\text{end}} \longleftarrow i_{\text{stt}} + K_r - 1$; $\boldsymbol{\omega} \longleftarrow$ entries i_{stt} to i_{end} of the mth column of $\boldsymbol{u}_{\text{SMC}}$

$i_{\text{stt}} \longleftarrow i_{\text{end}} + 1$

$(\boldsymbol{\sigma}^2_{u\text{SMC}})_{rm} \sim \text{Inverse-Gamma}\left(\frac{1}{2}(K_r+1), (\boldsymbol{a}_{u\text{SMC}})_{rm}^{-1} + \frac{1}{2}\|\boldsymbol{\omega}\|^2\right)$.

Produce summaries based on the current approximate posterior distributions of $\begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{u} \end{bmatrix}$, σ^2_ε and $(\sigma^2_{u1}, \ldots, \sigma^2_{uR})$ equalling the probability mass functions with atoms stored in $\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}$, $\boldsymbol{\sigma}^2_{\varepsilon\text{SMC}}$ $\boldsymbol{\sigma}^2_{u\text{SMC}}$ respectively, and probabilities $\boldsymbol{p}$.

until data no longer available or analysis terminated.

4.1 Generalized Linear Models

As with the (7) set-up, let $\boldsymbol{X}$ be a $n \times p$ design matrix and $\boldsymbol{\beta}$ be a $p \times 1$ coefficient vector. In this subsection we now suppose that the entries of the $n \times 1$ response vector $\boldsymbol{y}$ have the following one-parameter exponential family probability mass or density function:

$$\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta}) = \exp\left\{\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta} - \mathbf{1}^T b\big(\boldsymbol{X}\boldsymbol{\beta}\big) + \mathbf{1}^T c(\boldsymbol{y})\right\} h(\boldsymbol{y}) \tag{14}$$

for particular scalar-to-scalar functions b, c and h with the convention that function evaluation is applied in an element-wise fashion. The logistic regression special case, for binary response data, corresponds to

$$b(x) = \log(e^x + 1), \quad c(x) = 0 \quad \text{and} \quad h(x) = I(x \in \{0, 1\}). \tag{15}$$

Instead, setting

$$b(x) = e^x, \quad c(x) = -\log(x!) \quad \text{and} \quad h(x) = I(x \in \{0, 1, 2, \ldots\}) \tag{16}$$

corresponds to Poisson regression for count responses.

Online fitting of (14) is provided by Algorithm 6 and justified in Section S.2.6 of the supplement. An important difference between Algorithm 6 for generalized linear models and Algorithm 3 for Gaussian response linear models is that purely online fitting is *not* being achieved. Recall that Algorithm 3 is such that the data can be discarded after each sufficient statistic update is accomplished. In contrast, Algorithm 6 is such that, every time a new data vector arrives the full data to date needs to be available and processed for the approximate posterior distribution updates. The essence of this difference is the non-Gibbsian nature of the $\boldsymbol{\beta}$ vector full conditional distribution in the generalized response situation. Instead of the simple closed form update that arises in the Gaussian case, a Metropolis-Hastings scheme has to be called up. The logarithm of Metropolis-Hastings ratio, denoted in Algorithm 6 by λ, requires the full data to date.

4.1.1 Choice of the Metropolis-Hastings Random Walk Scale Parameter

Algorithm 6 involves the following step:

$$\boldsymbol{\beta}_{\text{RW}} \longleftarrow \boldsymbol{\beta}_{\text{SMC},m} + \frac{\upsilon \boldsymbol{z}}{\sqrt{n}} \tag{17}$$

for some choice of the scale parameter $\upsilon > 0$. As explained in Section S.2.6 of the supplement, (17) corresponds to drawing from random walk proposal distribution as part of a Metropolis-Hastings scheme for obtaining a draw from the current full conditional distribution of $\boldsymbol{\beta}$.

Several strategies to select υ have been developed. Sophisticated approaches such as the Metropolis-adjusted Langevin algorithm (Roberts & Stramer, 2003) introduce also a deterministic drift and exploit gradient information to provide principled choices of υ, which can be further refined by considering higher-order derivatives (Girolami & Calderhead, 2011). Fearnhead and Taylor (2013) were among the first to bring ideas from adaptive Markov chain Monte Carlo to bear in the context of sequential Monte Carlo, and suggested to adapt υ based on the expected squared jumping distance; see also Bon *et al.* (2021). Maximising the expected squared jumping distance is equivalent to minimising the first-order autocorrelation of the Markov chain and computation is straightforward. Chopin and Papaspiliopoulos (2020, Section 17.2.1) recommend to use an estimate of the sample covariance from the previous step of sequential Monte Carlo to calibrate the covariance matrix of a general multivariate Gaussian proposal. However, despite the potential for these approaches to deliver improved mixing, they can also introduce novel failure modes. For example, expected squared jumping distance may not be concave as υ is varied, which means that optimisation could, in principle, be difficult. To promote robustness, here we use a simpler approach, based on theoretical results in Roberts & Rosenthal (2001). This entails choosing υ so that about 23% of the particles are updated according to the Algorithm 6 step:

$$\text{if } \lambda > \log(u) \text{ then the } m\text{th column of } \boldsymbol{\beta}_{\text{SMC}} \longleftarrow \boldsymbol{\beta}_{\text{RW}} \ (m = 1, \ldots, M).$$

The "about 23% of particles updated" approach to setting υ values can be done using the warm-up phase, and also involve simple-to-implement adaptations to υ as the data stream in. The illustrations in Sections 5.1 and 5.2 use such an approach for choice of υ.

4.2 Generalized Linear Mixed Models

The class of Bayesian generalized linear mixed models which we consider is

$$\begin{gathered} \mathfrak{p}(\boldsymbol{y}|\,\boldsymbol{\beta}, \boldsymbol{u}) = \exp\left\{\boldsymbol{y}^T(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u}) - \mathbf{1}^T b(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u}) + \mathbf{1}^T c(\boldsymbol{y})\right\} h(\boldsymbol{y}) \\ \boldsymbol{u}|\,\sigma^2_{u1}, \ldots, \sigma^2_{uR} \sim N(\mathbf{0}, \text{blockdiag}(\sigma^2_{u1}\,\boldsymbol{I}_{K_1}, \ldots, \sigma^2_{uR}\,\boldsymbol{I}_{K_r})) \end{gathered} \tag{18}$$

where $\boldsymbol{\beta}$ and σ^2_{ur}, $1 \leq r \leq R$, have prior distributions as given by (11). Model (18) has similar utility to (10) for various semiparametric regression scenarios, but for generalized response situations. In particular, logistic mixed models and Poisson mixed models correspond to the b, c and h functions given by (15) and (16), respectively.

Algorithm 7 describes online fitting of (18) via sequential Monte Carlo, with justification provided by Section S.2.7 of the supplement. An illustration of Algorithm 7 is given in Section 5.2.

Algorithm 6 *Online sequential Monte Carlo algorithm for online approximate inference in the generalized response linear model.*

Tuning Parameter Inputs (defaults): $M \in \mathbb{N}\ (1000)$; $\tau > 0\ (2/M)$; $\upsilon > 0$ (see Section 4.1.1).

Hyperparameter Inputs: $\boldsymbol{\mu}_{\boldsymbol{\beta}}\ (p \times 1)$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}\ (p \times p)$ symmetric and positive definite.

Perform batch-based tuning runs analogous to those described in Algorithm 4 and determine a warm-up sample size n_{warm} for which convergence is validated.

Set $\boldsymbol{y}_{\text{warm}}$ and $\boldsymbol{X}_{\text{warm}}$ to be the response vector and design matrix based on the first n_{warm} observations. Then set $n \longleftarrow n_{\text{warm}}$, $\boldsymbol{y} \longleftarrow \boldsymbol{y}_{\text{warm}}$ and $\boldsymbol{X} \longleftarrow \boldsymbol{X}_{\text{warm}}$.

Cycle:

Read in $y_{\text{new}}(1 \times 1)$ and $\boldsymbol{x}_{\text{new}}(p \times 1)$; $n \longleftarrow n + 1$

$\boldsymbol{\eta} \longleftarrow \boldsymbol{\beta}_{\text{SMC}}^T \boldsymbol{x}_{\text{new}}$; $\boldsymbol{\ell} \longleftarrow \boldsymbol{\ell} + y_{\text{new}}\boldsymbol{\eta} - b(\boldsymbol{\eta})$

$\boldsymbol{p} \longleftarrow \exp\{\boldsymbol{\ell} - \max(\boldsymbol{\ell})\}/[\mathbf{1}^T \exp\{\boldsymbol{\ell} - \max(\boldsymbol{\ell})\}]$

If $\boldsymbol{p}^T\boldsymbol{p} > \tau$ then

$\boldsymbol{\beta}_{\text{SMC}} \longleftarrow \textsc{SystematicResample}\big(\boldsymbol{\beta}_{\text{SMC}}, \boldsymbol{p}\big)$; $\boldsymbol{\ell} \longleftarrow \log\big(\frac{1}{M}\big)\mathbf{1}$

$$\boldsymbol{y} \longleftarrow \begin{bmatrix} \boldsymbol{y} \\ y_{\text{new}} \end{bmatrix} ;\ \boldsymbol{X} \longleftarrow \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{x}_{\text{new}}^T \end{bmatrix}$$

For $m = 1, \ldots, M$:

$\boldsymbol{z} \longleftarrow p \times 1$ vector containing totally independent $N(0,1)$ draws

$\boldsymbol{\beta}_{\text{SMC},m} \longleftarrow m$th column of $\boldsymbol{\beta}_{\text{SMC}}$; $\boldsymbol{\beta}_{\text{RW}} \longleftarrow \boldsymbol{\beta}_{\text{SMC},m} + \dfrac{\upsilon \boldsymbol{z}}{\sqrt{n}}$

$\boldsymbol{\eta}_{\text{SMC}} \longleftarrow \boldsymbol{X}\boldsymbol{\beta}_{\text{SMC},m}$; $\boldsymbol{\eta}_{\text{RW}} \longleftarrow \boldsymbol{X}\boldsymbol{\beta}_{\text{RW}}$

$\lambda \longleftarrow \boldsymbol{y}^T(\boldsymbol{\eta}_{\text{RW}} - \boldsymbol{\eta}_{\text{SMC}}) - \mathbf{1}^T\{b(\boldsymbol{\eta}_{\text{RW}}) - b(\boldsymbol{\eta}_{\text{SMC}})\}$
$\quad -\frac{1}{2}\boldsymbol{\beta}_{\text{RW}}^T\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_{\text{RW}} + \frac{1}{2}\boldsymbol{\beta}_{\text{SMC},m}^T\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_{\text{SMC},m} + (\boldsymbol{\beta}_{\text{RW}} - \boldsymbol{\beta}_{\text{SMC},m})^T\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}}$

$u \longleftarrow$ draw from the Uniform$(0,1)$ distribution

if $\lambda > \log(u)$ then

mth column of $\boldsymbol{\beta}_{\text{SMC}} \longleftarrow \boldsymbol{\beta}_{\text{RW}}$

Produce summaries based on the current approximate posterior distribution of $\boldsymbol{\beta}$ equalling the probability mass functions with atoms stored in $\boldsymbol{\beta}_{\text{SMC}}$ and probabilities $\boldsymbol{p}$.

until data no longer available or analysis terminated.

Algorithm 7 *Online sequential Monte Carlo algorithm for approximate inference in the generalized linear mixed model (18).*

Tuning Parameter Inputs (defaults): $M \in \mathbb{N}\ (1000)$; $\tau > 0\ (2/M)$; υ (see Section 4.1.1).

Hyperparameter Inputs: $\boldsymbol{\mu}_{\boldsymbol{\beta}}\ (p \times 1)$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}\ (p \times p)$ symmetric and positive definite, $s_{ur} > 0$, $1 \leq r \leq R$.

Perform batch-based tuning runs analogous to those described in Algorithm 4 and determine a warm-up sample size n_{warm} for which convergence is validated.

Set $\boldsymbol{y}_{\text{warm}}$ and $\boldsymbol{C}_{\text{warm}}$ to be the response vector and design matrix based on the first n_{warm} observations.

Set the following matrices (with dimensions)

$$\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}\ (P \times M) \quad ; \quad \boldsymbol{\sigma}^2_{u\text{SMC}}\ (r \times M) \quad ; \quad \boldsymbol{a}_{u\text{SMC}}\ (r \times M)$$

such that each column is a random sample from the relevant approximate posterior distribution according to the batch Markov chain Monte Carlo samples based on the first n_{warm} observations.

Cycle:

Read in $y_{\text{new}}\ (1 \times 1)$ and $\boldsymbol{c}_{\text{new}}\ (P \times 1)$; $n \leftarrow n + 1$

$\boldsymbol{\eta} \longleftarrow \begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}^T \boldsymbol{c}_{\text{new}}$; $\boldsymbol{\ell} \longleftarrow \boldsymbol{\ell} + y_{\text{new}}\boldsymbol{\eta} - b(\boldsymbol{\eta})$

$\boldsymbol{p} \longleftarrow \exp\{\boldsymbol{\ell} - \max(\boldsymbol{\ell})\}/[\mathbf{1}^T \exp\{\boldsymbol{\ell} - \max(\boldsymbol{\ell})\}]$

If $\boldsymbol{p}^T\boldsymbol{p} > \tau$ then

$$\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \\ \boldsymbol{\sigma}^2_{u\text{SMC}} \\ \boldsymbol{a}_{u\text{SMC}} \end{bmatrix} \longleftarrow \text{SYSTEMATICRESAMPLE}\left(\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \\ \boldsymbol{\sigma}^2_{u\text{SMC}} \\ \boldsymbol{a}_{u\text{SMC}} \end{bmatrix}, \boldsymbol{p} \right) ; \boldsymbol{\ell} \longleftarrow \log\left(\tfrac{1}{M}\right)\mathbf{1}$$

$$\boldsymbol{y} \longleftarrow \begin{bmatrix} \boldsymbol{y} \\ y_{\text{new}} \end{bmatrix} ; \boldsymbol{C} \longleftarrow \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{c}^T_{\text{new}} \end{bmatrix}$$

For $m = 1, \ldots, M$:

$\boldsymbol{z} \longleftarrow P \times 1$ vector containing totally independent $N(0,1)$ draws

$$\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}_m \longleftarrow m\text{th column of } \begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix} ; \begin{bmatrix} \boldsymbol{\beta}_{\text{RW}} \\ \boldsymbol{u}_{\text{RW}} \end{bmatrix} \longleftarrow \begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}_m + \frac{\upsilon \boldsymbol{z}}{\sqrt{n}}$$

$$\boldsymbol{\eta}_{\text{SMC}} \longleftarrow \boldsymbol{C}\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}_m ; \boldsymbol{\eta}_{\text{RW}} \longleftarrow \boldsymbol{C}\begin{bmatrix} \boldsymbol{\beta}_{\text{RW}} \\ \boldsymbol{u}_{\text{RW}} \end{bmatrix}$$

$$\lambda \longleftarrow \boldsymbol{y}^T(\boldsymbol{\eta}_{\text{RW}} - \boldsymbol{\eta}_{\text{SMC}}) - \mathbf{1}^T\{b(\boldsymbol{\eta}_{\text{RW}}) - b(\boldsymbol{\eta}_{\text{SMC}})\}$$
$$-\tfrac{1}{2}\boldsymbol{\beta}^T_{\text{RW}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\beta}}\boldsymbol{\beta}_{\text{RW}} + \tfrac{1}{2}\boldsymbol{\beta}^T_{\text{SMC},m}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\beta}}\boldsymbol{\beta}_{\text{SMC},m} + (\boldsymbol{\beta}_{\text{RW}} - \boldsymbol{\beta}_{\text{SMC},m})^T\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\beta}}\boldsymbol{\mu}_{\boldsymbol{\beta}}$$

$i_{\text{stt}} \longleftarrow 1$

For $r = 1, \ldots, R$:

$i_{\text{end}} \longleftarrow i_{\text{stt}} + K_r - 1$;

$\boldsymbol{\omega}_{\text{SMC}} \longleftarrow$ entries i_{stt} to i_{end} of the mth column of $\boldsymbol{u}_{\text{SMC}}$

$\boldsymbol{\omega}_{\text{RW}} \longleftarrow$ entries i_{stt} to i_{end} of the mth column of $\boldsymbol{u}_{\text{RW}}$

continued on a subsequent page ...

Algorithm 7 continued. *This is a continuation of the description of this algorithm that continues on a preceding page.*

$i_{\text{stt}} \longleftarrow i_{\text{end}} + 1$

$\lambda \longleftarrow \lambda - \frac{1}{2}\big(\|\boldsymbol{\omega}_{\text{SMC}}\|^2 - \|\boldsymbol{\omega}_{\text{RW}}\|^2\big) \Big/ (\boldsymbol{\sigma}^2_{u\text{SMC}})_{rm}$

$u \longleftarrow$ draw from the Uniform$(0, 1)$ distribution

if $\lambda > \log(u)$ then

mth column of $\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix} \longleftarrow \begin{bmatrix} \boldsymbol{\beta}_{\text{RW}} \\ \boldsymbol{u}_{\text{RW}} \end{bmatrix}$

$i_{\text{stt}} \longleftarrow 1$

For $r = 1, \ldots, R$:

$(\boldsymbol{a}_{u\text{SMC}})_{rm} \sim$ Inverse-Gamma $\big(1, (\boldsymbol{\sigma}^2_{u\text{SMC}})^{-1}_{rm} + s^{-1}_{ur}\big)$

$i_{\text{end}} \longleftarrow i_{\text{stt}} + K_r - 1$; $\boldsymbol{\omega} \longleftarrow$ entries i_{stt} to i_{end} of the mth column of $\boldsymbol{u}_{\text{SMC}}$

$i_{\text{stt}} \longleftarrow i_{\text{end}} + 1$

$(\boldsymbol{\sigma}^2_{u\text{SMC}})_{rm} \sim$ Inverse-Gamma $\big(\frac{1}{2}(K_r + 1), (\boldsymbol{a}_{u\text{SMC}})^{-1}_{rm} + \frac{1}{2}\|\boldsymbol{\omega}\|^2\big)$.

Produce summaries based on the current approximate posterior distributions of $\begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{u} \end{bmatrix}$ and $(\sigma^2_{u1}, \ldots, \sigma^2_{uR})$ equalling the probability mass functions with atoms stored in $\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}$ and $\boldsymbol{\sigma}^2_{u\text{SMC}}$ respectively, and probabilities $\boldsymbol{p}$.

until data no longer available or analysis terminated.

5 Illustrations

We have tested Algorithms 3–7 on many simulated and actual data sets. In this section we give some illustrations of the practical performance of the new methodology. The first one includes a comparison with the Luts *et al.* (2014) variational approach.

5.1 Online Logistic Regression

Algorithm 5 of Luts *et al.* (2014) offers real-time logistic regression with pure online updating based on the logistic log-likelihood variational approximations of Jaakkola & Jordan (2000). However, as we mentioned in Section 1, this online mean field variational Bayes approach to logistic regression is susceptible to poor accuracy. This problem, and the online sequential Monte Carlo remedy, is illustrated here via a simple linear logistic regression scenario.

Suppose that new predictor/response pairs $(x_{\text{new}}, y_{\text{new}})$ are generated according to

$$x_{\text{new}} \sim \text{Uniform}(0, 1), \quad y_{\text{new}}|x_{\text{new}} \sim \text{Bernoulli}\big(\text{expit}(\beta_{0,\text{true}} + \beta_{1,\text{true}}\, x_{\text{new}})\big) \tag{19}$$

where $\beta_{0,\text{true}} = -7.5$ and $\beta_{1,\text{true}} = 9.36$. Of interest here are the posterior density functions of the coefficient parameters

$$\mathfrak{p}(\beta_0|\boldsymbol{x}_{\text{curr}}, \boldsymbol{y}_{\text{curr}}) \quad \text{and} \quad \mathfrak{p}(\beta_1|\boldsymbol{x}_{\text{curr}}, \boldsymbol{y}_{\text{curr}})$$

where $\boldsymbol{x}_{\text{curr}}$ and $\boldsymbol{y}_{\text{curr}}$ are the current predictor and response vectors as the data stream in according to

$$n \longleftarrow n + 1, \quad \boldsymbol{x}_{\text{curr}} \longleftarrow (\boldsymbol{x}_{\text{curr}}, x_{\text{new}}) \quad \text{and} \quad \boldsymbol{y}_{\text{curr}} \longleftarrow (\boldsymbol{y}_{\text{curr}}, y_{\text{new}}).$$

We warmed up both Algorithm 6 of the present paper and Algorithm 5 of Luts *et al.* (2014) with a sample size of $n_{\text{warm}} = 100$ and terminated at $n = 500$. As a "gold standard" we also

obtain the batch Monte Carlo Markov chain fits to each of the $n = 100, 101, \ldots, 500$ data sets using the package rstan (Guo *et al.*, 2023) within the R computing environment (R Core Team, 2024). A movie in the supplemental material [1] of this article displays and compares the approximations of $\mathfrak{p}(\beta_0|\boldsymbol{x}_{\text{curr}}, \boldsymbol{y}_{\text{curr}})$ and $\mathfrak{p}(\beta_1|\boldsymbol{x}_{\text{curr}}, \boldsymbol{y}_{\text{curr}})$. Figure 3 shows four frames of the movie for $n \in \{200, 300, 400, 500\}$ and the parameter β_1. The approximations to the $\mathfrak{p}(\beta_j|\boldsymbol{x}_{\text{curr}}, \boldsymbol{y}_{\text{curr}})$ based on batch Markov chain Monte Carlo and online sequential Monte Carlo are displayed using frequency polygons, as described in Section S.2.8 of the supplement with bin width rule (S.11). The online mean field variational Bayes approximations are Normal density functions.

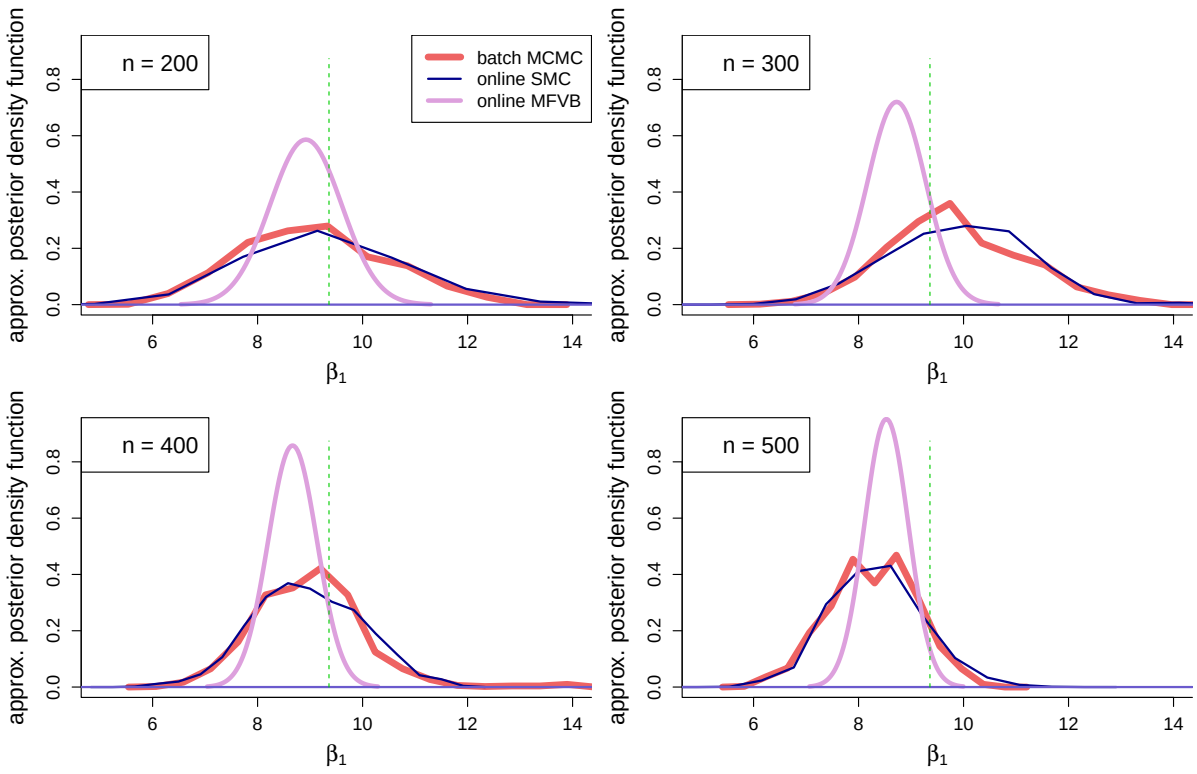

Figure 3: *Comparison of the approximations of* $\mathfrak{p}(\beta_1|\boldsymbol{x}_{curr}, \boldsymbol{y}_{curr})$ *for the online logistic regression example. The batch Markov chain Monte Carlo (MCMC) approximation is displayed as a frequency polygon density estimate based on a kept MCMC sample of size* $1,000$ *and bin width given by (S.11). The online sequential Monte Carlo (SMC) approximation is displayed as a frequency polygon representation of a probability mass function with* $1,000$ *atoms, as defined in Section S.2.8 of the supplement, and same bin width as the MCMC frequency polygon. The online mean field variational Bayes (MFVB) approximations, corresponding to Algorithm 5 of Luts et al. (2014), are Normal density functions. The dashed vertical line corresponds to* $\beta_{1,true} = 9.36$.

Figure 3 and its extended movie form show that online sequential Monte Carlo leads to very good approximation of the posterior distributions of β_0 and β_1. In contrast, online mean field variational Bayes provides overly narrow posterior density function approximations for this example.

A price to be paid for sequential Monte Carlo's improved accuracy is slower computing time. Figure 4 conveys this cost when running the relevant algorithms in the R computing environment (R Core Team, 2024) on the third author's MacBook Air laptop, which has a 3.2 gigahertz processor and 16 gigabytes of random access memory. The jaggedness in the Figure 4 curves is due to rounding. It takes the sequential Monte Carlo about 2.75 seconds to get from $n = 100$ to $n = 500$ whereas mean field variational Bayes takes only 0.01 second. The Figure 4 curves show the times and their ratios for getting from $n = 100$ to intermediate sample sizes. The sequential Monte Carlo computing times appear to be quadratic in sample size whilst the

[1] Currently the movie is on the web-page: `http://matt-p-wand.net/MOWmovies.html`

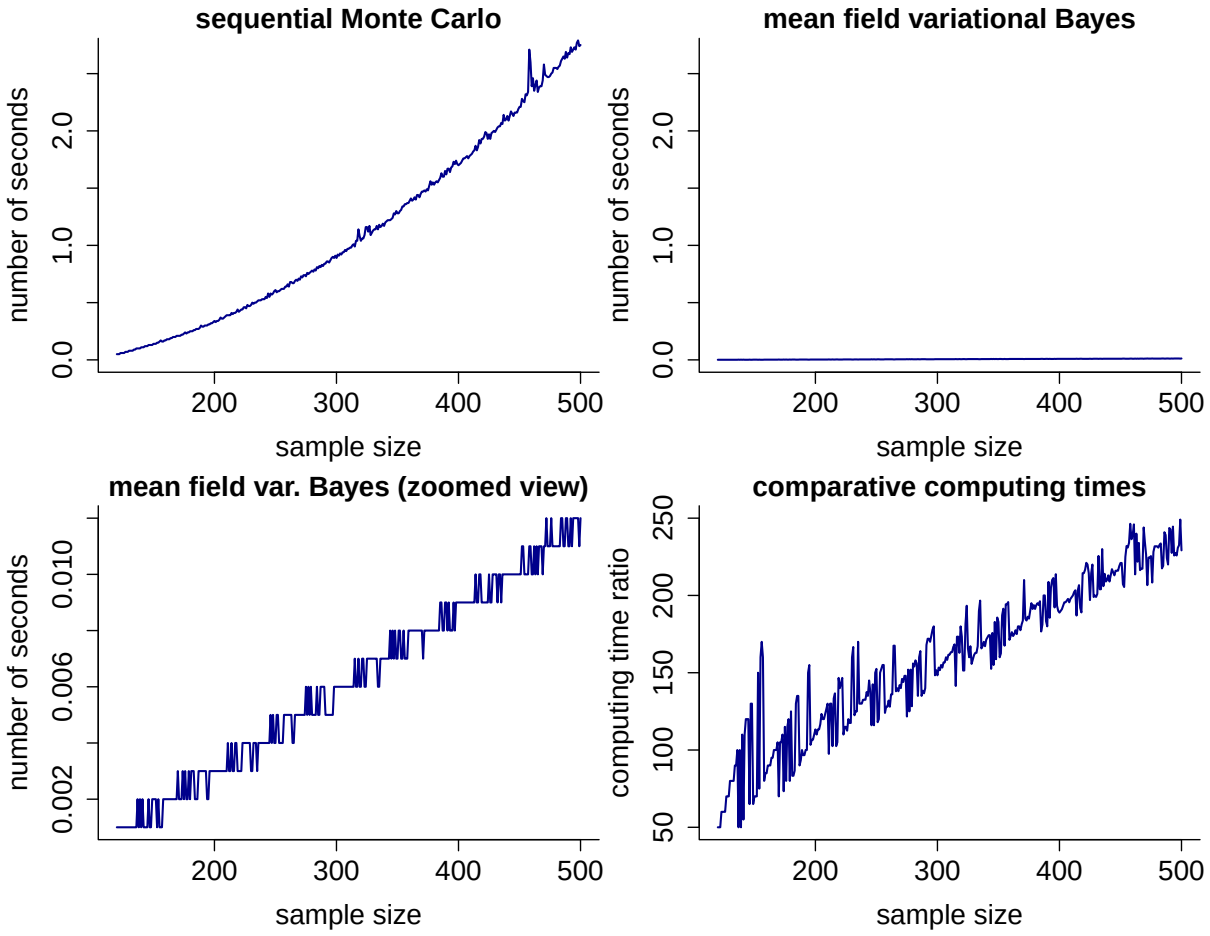

Figure 4: *The computing times and their ratios for the Figure 3 example. In each panel the horizontal axis corresponds to the sample size in the online fitting phase. In the upper panels the vertical axis corresponds to number of seconds required for online fitting and inference after the warm-up phase and the axis limits are the same to aid visual comparison. The lower left panel is a zoomed view of the upper right panel's curve. The vertical axis in the lower right panel corresponds to computing time ratios.*

mean field variational Bayes times are linear. The "lightning fast" aspect of the online mean field variational Bayes needs to be traded off against it being prone to inaccurate inference, as Figure 3 demonstrates.

5.2 Online Binary Response Nonparametric Regression

This second simulated data illustration involves extension of (19) to

$$x_{\text{new}} \sim \text{Uniform}(0,1), \quad y_{\text{new}}|x_{\text{new}} \sim \text{Bernoulli}\big(f_{\text{true}}(x_{\text{new}})\big) \tag{20}$$

for a smooth function f_{true} such that $0 \leq f_{\text{true}}(x) \leq 1$ for $x \in (0,1)$. In this section's illustrations we have

$$f_{\text{true}}(x) \equiv \{1.05 - 1.02\,x + 0.018\,x^2 + 0.4\,\phi(x; 0.38, 0.08) + 0.08\,\phi(x; 0.75, 0.03)\}/2.7$$

where $\phi(\cdot; \mu, \sigma)$ denotes the density function of the $N(\mu, \sigma^2)$ distribution.

Online estimation and inference concerning f_{true} can be achieved using the $R = 1$ version of Algorithm 7 with the current $\boldsymbol{X}$ and $\boldsymbol{Z}$ matrices set to

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \quad \text{and} \quad \boldsymbol{Z} = \begin{bmatrix} z_1(x_1) & \cdots & z_K(x_1) \\ \vdots & \ddots & \vdots \\ z_1(x_n) & \cdots & z_K(x_n) \end{bmatrix} \tag{21}$$

where $\{z_k(\cdot) : 1 \leq k \leq K\}$ is a suitable spline basis such as that described in Section 4 of Wand & Ormerod (2008). The $\boldsymbol{C}$ matrix appearing in Algorithm 7 is then given by $\boldsymbol{C} = [\boldsymbol{X}\ \boldsymbol{Z}]$. For this example we have $K = 37$.

With the online sequential Monte Carlo particles having a dimension of around 40 we found that a substantial batch-based warm-up was required. This aspect is conveyed by Figure 5 which compares the diagnostic plots corresponding to the generalized linear mixed model extension of Algorithm 4 for $n_{\text{warm}} = 100$ and $n_{\text{warm}} = 500$. It is apparent from Figure 5 that the lower warm-up sample size is not adequate and one around five times larger is desirable for good online estimation and inference for f_{true}. An analogous phenomenon was observed for the online mean field variational Bayes approach used by Luts *et al.* (2014), with Figure 5 of that article showing that $n_{\text{warm}} = 100$ is inadequate for a similar model. Since the updates occur in a high-dimensional Euclidean space, and the mixing properties of random walk Metropolis–Hastings algorithms deteriorate in a manner inversely proportional to the dimension of the distribution being sampled (Gelman, Gilks & Roberts, 1997), the starting values corresponding to lower n_{warm} are more susceptible to divergence away from the correct posterior distributions.

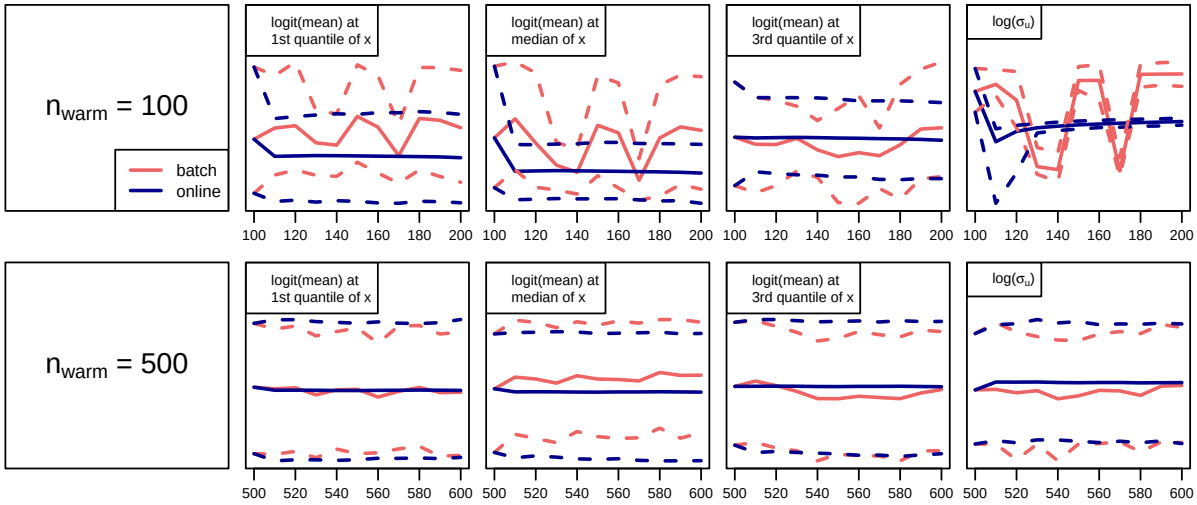

Figure 5: *Batch-based convergence diagnostics for the binary response nonparametric regression example. The solid lines track posterior means, while the dashed lines show the limits of corresponding 95% credible intervals. First row: the horizontal axes correspond to sample sizes between a warm-up batch sample of size $n_{warm} = 100$ and validation samples up to $n_{valid} = 100$ greater than n_{warm}. Second row: as for the first row, but with $n_{warm} = 500$.*

A movie in the supplemental material [2] of this article displays and compares the online sequential Monte Carlo and batch Markov chain Monte Carlo estimates, and variability, bands of f_{true}. Figure 6 displays four frames of this movie, for the sample sizes $n \in \{1000, 2000, 3500, 5000\}$. There is very good correspondence between the online and batch fits.

5.3 Illustration for Actual Data

To illustrate the methodology on actual data we applied Algorithm 5 for online additive model fitting to sequentially arriving data from the data frame `SydneyRealEstate` within the R package HRW (Harezlak *et al.*, 2021). The data consist of numerical attributes of 37,676 houses sold in Sydney, Australia, during 2001. The response variable is the natural logarithm of sale price in Australian dollars. After conversion of categorical variables to indicator variables there are around 40 candidate predictors. Most of the candidate predictors are continuous and could impact the mean response either linearly or non-linearly. Given that this is just an

[2] Currently the movie is on the web-page: `http://matt-p-wand.net/MOWmovies.html`

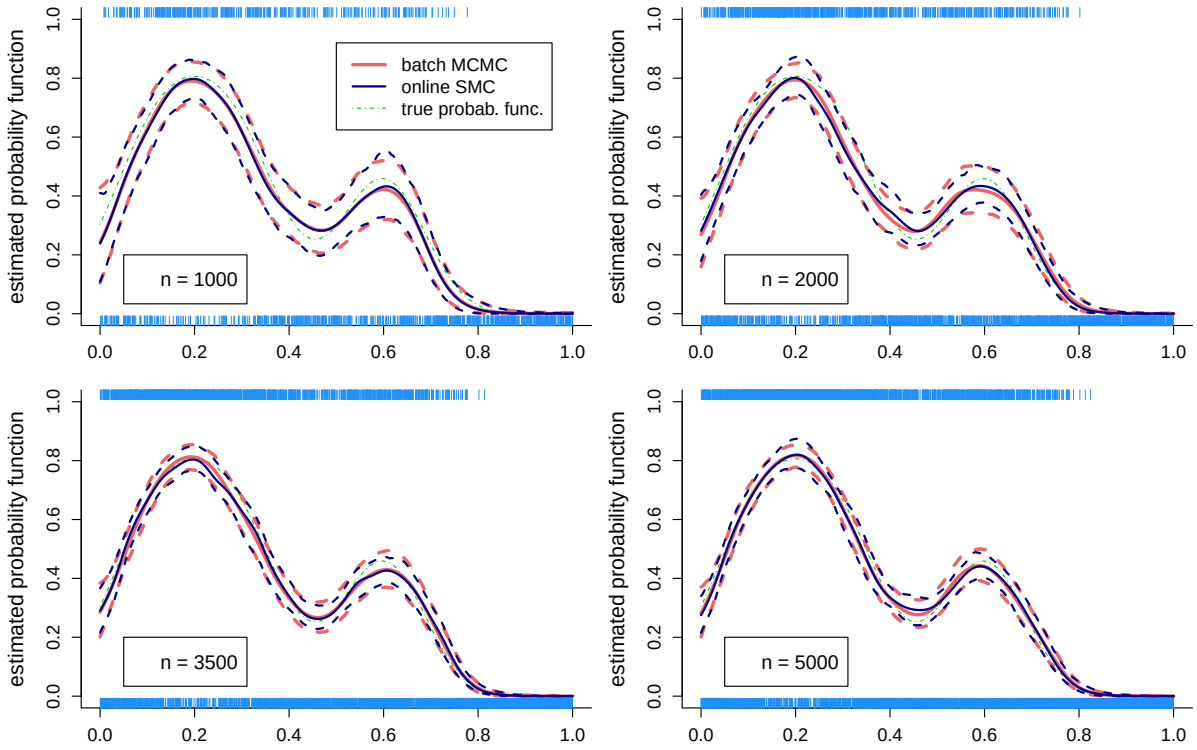

Figure 6: *Comparison of online sequential Monte Carlo and batch Markov chain Monte Carlo inference for the probability function f_{true} in the binary response nonparametric regression example for four example sample sizes from the movie in the supplemental material. The solid curves correspond to the posterior mean, which is targeting the true probability function shown as a dot-dashed curve. The dashed curves correspond to pointwise 95% credible intervals.*

illustration, we first ran the full data through the generalized additive model selection procedure provided by the R package gamselBayes (He & Wand, 2023) and arrived at 12 predictors entering the model linearly and 14 predictors entering the model non-linearly. Table 1 describes each of these 26 predictors. Fuller details are provided in the documentation of `SydneyRealEstate` within the HRW package. For the Bayesian model fitting all variables were linearly transformed to the unit interval and approximate non-informativity was imposed through the use of the hyperparameter choices $\boldsymbol{\mu}_{\boldsymbol{\beta}} = \mathbf{0}$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = 10^{10}\boldsymbol{I}$ and $s_\varepsilon = s_{ur} = 10^5$. For the upcoming graphical summaries all posterior distributions were back-transformed to correspond to the original units. The spline basis functions for each predictor are analogous to those given by (21) but with $K = 17$.

Algorithm 5 was warmed up with a batch Markov chain Monte Carlo fit to the first $n = 1000$ fields of `SydneyRealEstate`. This was followed by online additive model updating for sequentially arriving data based on the next 4000. For comparison, we then obtained the batch fits for each of the $n \in \{1000, 1010, 1020, \ldots, 5000\}$ data sets. A movie in the supplemental material [3] shows the online additive model fits and compares them with their batch counterparts. Figure 7 shows the $n = 3000$ frame of the movie. In both the movie and Figure 7, the approximate posterior density functions of the coefficients for the linear effect predictors are displayed using frequency polygons as described in Section S.2.8 of the supplement. The non-linear effect plots correspond to slices of fitted surface with all other predictors set to their median values. The solid curves show posterior means and the dashed curves show pointwise 95% credible intervals. Both the movie and Figure 7 demonstrate online Bayesian inference via Algorithm 5 essentially matching the results from successive batch analyses.

[3] Currently the movie is on the web-page: `http://matt-p-wand.net/MOWmovies.html`

predictors entering linearly	predictors entering non-linearly
degrees longitude	lot size
distance to nearest highway	degrees latitude
distance to harbour tunnel	inflation rate measure
nitric oxide level	average income of suburb
suspended matter level	distance to nearest bus stop
ozone level	distance to nearest park
particulate matter < 10 micrometers	distance to nearest main road
sodium dioxide level	distance to nearest sealed road
distance to nearest medical services	distance to nearest unsealed road
indicator sale in 2nd quarter	proportion of foreigners in suburb
indicator sale in 3rd quarter	distance to nearest ambulance
indicator sale in 4th quarter	distance to nearest factory
	distance to nearest hospital
	distance to nearest school

Table 1: *Predictors used in the online additive model fitting illustration for real estate data for houses sold in Sydney, Australia, during 2001.*

6 Concluding Remarks

We have demonstrated that sequential Monte Carlo provides a viable approach to online, or real-time, semiparametric regression that overcome the accuracy shortcomings of the mean field variational Bayes approach. Our algorithms facilitate straightforward implementation in a wide range of semiparametric regression settings. For generalized response models and particular applications, some modifications concerning the Metropolis-Hastings steps may be worth considering. We have provided the foundations upon which such modifications could be carried out. Extensions to more elaborate models can also be entertained, with the current article as a solid basis.

7 Acknowledgements

We are grateful to David Leslie and Matt McLean for their contributions to this research. We also acknowledge helpful reviewer comments. This work was funded by Australian Research Council Discovery Project DP140100441.

References

Bon, J.J., Lee, A. & Drovandi, C. (2021). Accelerating sequential Monte Carlo with surrogate likelihoods. *Statistics and Computing*, **31**, Article number 62.

Carroll, R.J. (1976). On sequential density estimation. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, **36**, 137–151.

Chopin, N. & Papaspiliopoulos, O. (2020). *An Introduction to Sequential Monte Carlo*. Cham, Switzerland: Springer.

Del Moral, P., Doucet, A. & Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B*, **68**, 411–436.

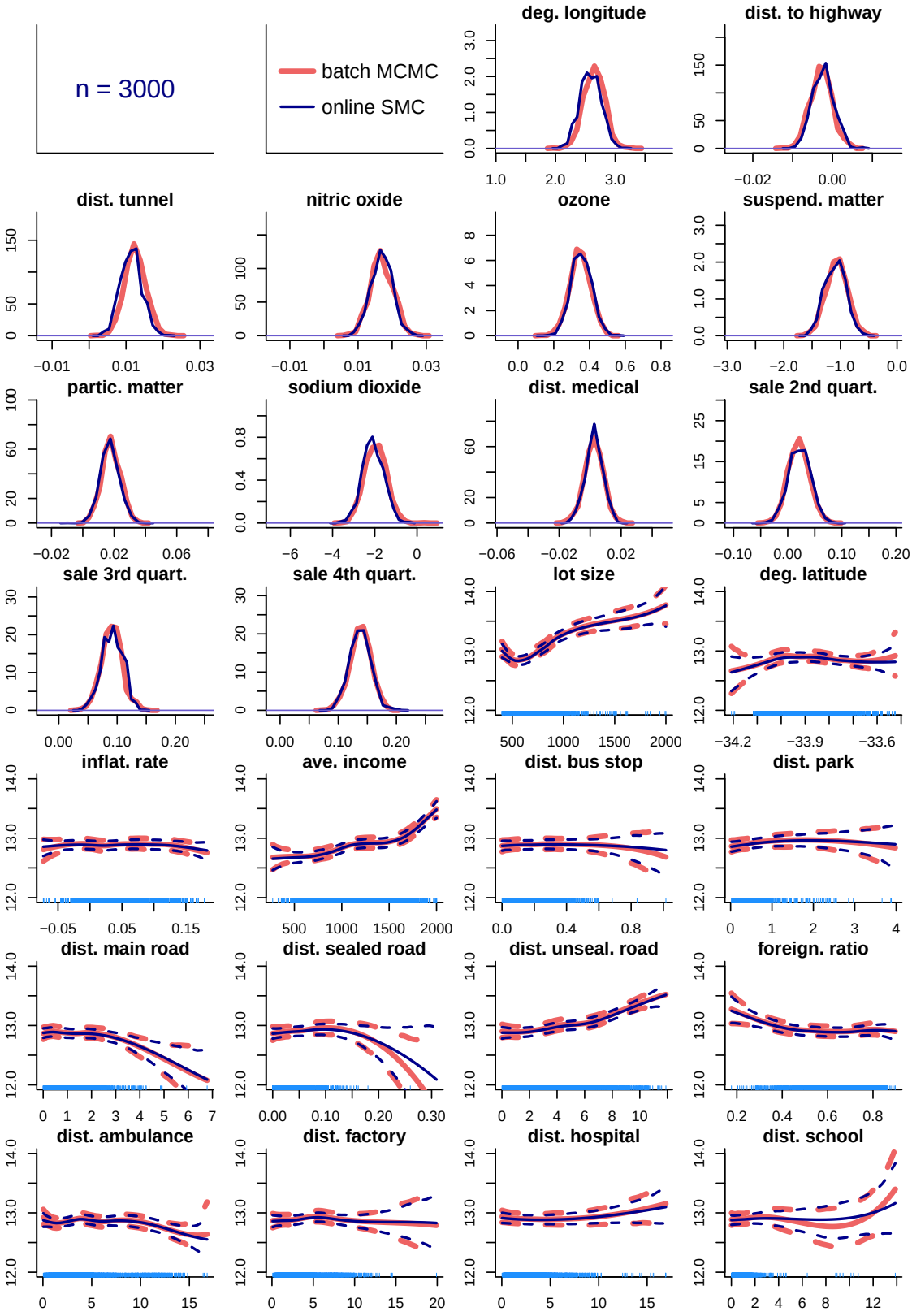

Figure 7: *Comparison of online sequential Monte Carlo and batch Markov chain Monte Carlo inference for the additive model fits to the Sydney real estate data for a sample size of* $n = 3000$. *The frequency polygon plots correspond to approximate posterior density functions of the coefficients for predictors entering the model linearly. The subsequent panels show the estimates of the nonlinear effects for predictors entering the model nonlinearly. The solid curves are posterior means, with each other predictor set to its median value. The dashed curves are pointwise 95% credible intervals.*

Fearnhead, P. & Taylor, B.M. (2013). An adaptive sequential Monte Carlo sampler. *Bayesian Analysis*, **8**, 411–438.

Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A. & Rubin, D.B. (2014). *Bayesian Data Analysis, Third Edition*, Boca Raton, Florida: CRC Press.

Gelman, A., Gilks, W., Roberts, G. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability*, **7**, 110—120.

Gilks, W.R. & Berzuini, C. (2001). Following a moving target – Monte Carlo inference for dynamic models. *Journal of the Royal Statistical Society, Series B*, **63**, 127–146.

Girolami, M. & Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods (with discussion). *Journal of the Royal Statistical Society, Series B*, **73**, 123–214.

Gramacy, R.B. & Polson, N.G. (2011). Particle learning of Gaussian process models for sequential design and optimization. *Journal of Computational and Graphical Statistics*, **20**, 102–118.

Guo, J., Gabry, J., Goodrich, B. & Weber, S. (2023). rstan: R interface to Stan. R package version 2.21.8. `https://cran.r-project.org`

Harezlak, J., Ruppert, D. & Wand, M.P. (2018). *Semiparametric Regression with R*. New York: Springer.

Harezlak, J., Ruppert, D. & Wand, M.P. (2021). HRW. Datasets, functions and scripts for semiparametric regression supporting Harezlak, Ruppert & Wand (2018). R package version 1.0. `https://CRAN.R-project.org`

He, V.X. & Wand, M.P. (2023). gamselBayes: Bayesian generalized additive model selection. R package version 2.0. `https://cran.r-project.org`

Huang, A. & Wand, M.P. (2013). Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis*, **8**, 439–452.

Jaakkola, T.S. & Jordan, M.I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing* **10**, 25–37.

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

Krzyzak, A. & Pawlak, M. (1982). Almost everywhere convergence of recursive kernel regression function estimates. In *Probability and Statistical Inference: Proceedings of the 2nd Pannonian Symposium on Mathematical Statistics*, editors: W. Grossman, G.C. Pflug & W. Wertz, pp. 191–209.

Liu, J.S. & Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, **93**, 1032–1044.

Luo, L & Song, P.X.K. (2023). Multivariate online regression analysis with heterogeneous streaming data. *Canadian Journal of Statistics*, **51**, 111–133.

Luts, J., Broderick, T. & Wand, M.P. (2014). Real-time semiparametric regression. *Journal of Computational and Graphical Statistics*, **23**, 589–615.

Pitt, M. & Shephard, N. (1999). Filtering via simulation: auxiliary particle filters. *Journal of Computational and Graphical Statistics*, **94**, 590–599.

Plummer, M. (2022). rjags: Bayesian graphical models using Markov chain Monte Carlo. R package version 4-13. https://cran.r-project.org/package=rjags

R Core Team (2024). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

Roberts, G.O. & Rosenthal, J.S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, **16**, 351–367.

Roberts, G.O. & Stramer, O. (2003). Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and Computing in Applied Probability*, **4**, 337–358.

Ruppert, D., Wand, M.P. & Carroll, R.J. (2003). *Semiparametric Regression*. New York: Cambridge University Press.

Stan Development Team (2022). Stan Modeling Language Users Guide and Reference Manual. Version 2.30. https://mc-stan.org

Talagala, P.D. Hyndman, R.J., Smith-Miles, K., Kandanaarachchi, S. & Muñoz, M.A. (2020). Anomaly detection in streaming non-stationary temporal data. *Journal of Computational and Graphical Statistics*, **29**, 13–27.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, **22**, 1701–1728.

Wand, M.P. & Ormerod, J.T. (2008). On semiparametric regression with O'Sullivan penalized splines. *Australian and New Zealand Journal of Statistics*, **50**, 179–198.

Weng, R.C.-H. & Coad, D.S. (2018). Real-time Bayesian parameter estimation for item response models. *Bayesian Analysis*, **13**, 115–137.

Yamato, H. (1971). Sequential estimation of a continuous probability density function and mode. *Bulletin of Mathematical Statistics*, **14**, 1–12.

Yin, G.G. & Yin, K. (1996). Passive stochastic approximation with constant step size and window width. *IEEE Transactions on Automatic Control*, **41**, 90–106.

Supplement for:

Online Semiparametric Regression via Sequential Monte Carlo

MARIANNE MENICTAS[1], CHRIS J. OATES[2] & MATT P. WAND[3]

[1]*Grubhub Inc., U.S.A.,* [2]*Newcastle University, U.K.*
and [3]*University of Technology Sydney, Australia*

S.1 Sequential Monte Carlo Details

We now provide details on the sequential Monte Carlo approach to online semiparametric regression described in Figure 2. Note, however, that the approach is quite generic and holds for wide classes of Bayesian graphical models. Our presentation is at the generic level.

Our description builds up to the final algorithm in stages. We start with two simplistic versions of sequential Monte Carlo for online fitting which omit the resampling enhancement provided by the SYSTEMATICRESAMPLE algorithm. This temporary omission allows the essence of the approach to be explained more straightforwardly.

S.1.1 Simplistic Versions of Sequential Monte Carlo

For each of the semiparametric regression models of interest, the starting situation of no data observed corresponds to $n = 0$. The arrival of the first response observation y_1, and its corresponding predictor vector, leads incrementation of the sample size to $n = 1$. Therefore, in terms of the response data, we have:

sample size:	$n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$\cdots$
new response:		y_1	y_2	y_3	y_4	$\cdots$

Each of our semiparametric regression models are such that, given the full set of parameters $\boldsymbol{\theta}$, the responses are conditionally independent. Hence, for any $n \in \mathbb{N}$:

$$\mathfrak{p}(y_1, y_2, \ldots, y_n|\boldsymbol{\theta}) = \mathfrak{p}(y_1|\boldsymbol{\theta})\mathfrak{p}(y_2|\boldsymbol{\theta}) \cdots \mathfrak{p}(y_n|\boldsymbol{\theta}). \tag{S.1}$$

S.1.1.1 Fixed Particle Case

The fixed particle case corresponds to the situation where the atoms of the probability mass function approximations of the posterior density functions

$$\mathfrak{p}(\boldsymbol{\theta}|y_1),\ \mathfrak{p}(\boldsymbol{\theta}|y_1, y_2),\ \mathfrak{p}(\boldsymbol{\theta}|y_1, y_2, y_3),\ \mathfrak{p}(\boldsymbol{\theta}|y_1, y_2, y_3, y_4), \ldots$$

are held fixed as the data stream in. At this point we stress that keeping the atoms fixed is not recommended in practice. We only do it here with pedagogy in mind. Denote the atoms by

$$\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M. \tag{S.2}$$

Let n be the current sample size and $\boldsymbol{y}_n$ be the corresponding vector of response values. The natural probability vector to put on the atoms is

$$\boldsymbol{p}^{[n]} \equiv \Big(\mathfrak{p}(\boldsymbol{\theta}_1|\boldsymbol{y}_n), \ldots, \mathfrak{p}(\boldsymbol{\theta}_M|\boldsymbol{y}_n)\Big) \Big/ \sum_{m=1}^{M} \mathfrak{p}(\boldsymbol{\theta}_m|\boldsymbol{y}_n).$$

Suppose that a new response observation y_{new} arrives so that the sample size becomes $n+1$ and the response vector becomes $\boldsymbol{y}_{n+1} \equiv (\boldsymbol{y}_n, y_{\text{new}})$. For the probability mass approximation of $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y}_{n+1})$ the natural probability vector to put on the atoms is

$$\boldsymbol{p}^{[n+1]} \equiv \Big(\mathfrak{p}(\boldsymbol{\theta}_1|\boldsymbol{y}_{n+1}), \ldots, \mathfrak{p}(\boldsymbol{\theta}_M|\boldsymbol{y}_{n+1})\Big) \Big/ \sum_{m=1}^{M} \mathfrak{p}(\boldsymbol{\theta}_m|\boldsymbol{y}_{n+1}).$$

In view of (S.1), as a function of $m \in \{1, \ldots, M\}$, we have

$$\frac{\boldsymbol{p}_m^{[n+1]}}{\boldsymbol{p}_m^{[n]}} \propto \frac{\mathfrak{p}(\boldsymbol{\theta}_m|\boldsymbol{y}_{n+1})}{\mathfrak{p}(\boldsymbol{\theta}_m|\boldsymbol{y}_n)} = \frac{\mathfrak{p}(\boldsymbol{y}_{n+1}|\boldsymbol{\theta}_m)\mathfrak{p}(\boldsymbol{\theta}_m)}{\mathfrak{p}(\boldsymbol{y}_n|\boldsymbol{\theta}_m)\mathfrak{p}(\boldsymbol{\theta}_m)} = \mathfrak{p}(y_{\text{new}}|\boldsymbol{\theta}_m).$$

Therefore, if the probability vector is initialized as $\boldsymbol{p}_m^{[0]} = 1/M$, the update steps

$$\boldsymbol{p}_m \longleftarrow \boldsymbol{p}_m \mathfrak{p}(y_{\text{new}}|\boldsymbol{\theta}_m) \quad ; \quad \boldsymbol{p}_m \longleftarrow \boldsymbol{p}_m \Big/ \sum_{m'=1}^{M} \boldsymbol{p}_{m'} \tag{S.3}$$

lead to the required probability mass function approximations as the data stream in.

The directed acyclic graphical nature of semiparametric regression models means that (S.3) often can be reduced to a simpler expression, such that $\boldsymbol{\theta}_m$ is replaced by the particle subvectors corresponding to the parents of $\boldsymbol{y}_n$. As an example, for the logistic additive model given by (4) and with directed acyclic graph representation in Figure 1 the update multiplicative factor simplifies to $\mathfrak{p}\big(y_{\text{new}}|\boldsymbol{\beta}_m, (\boldsymbol{u}_1)_m, (\boldsymbol{u}_2)_m\big)$.

Holding the particles to be fixed throughout the online analysis is far from desirable. Improved probability mass function approximation would be realised by having the particles reflect the ever-changing location, spread and shape of $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y}_n)$ as n increases. Section S.1.1.2 discusses one established strategy for varying the particles.

S.1.1.2 Varying Particles Adjustment

Several strategies have been developed for varying the particles after the arrival of a new observation vector. The one which we adopt for online semiparametric regression is known as the *resample-move* algorithm. Its description and justification are given in Section 3 of Gilks & Berzuini (2001), and is based on similar ideas from e.g. Kong *et al.* (1994). The algorithm involves both *resample* and *move* steps, which we now describe in turn.

The *resample* step involves drawing a sample of size M from the probability mass function with atoms corresponding to the current particles and probabilities corresponding to the current probability vector. It is motivated by circumvention of a problem known as *degeneracy*, which is a tendency for probability mass to be confined to a small number of particles as the iterations progress. The SystematicResample algorithm facilitates the resample step, using systematic resampling (see Section S.2.1 for details), for particles stored in matrix form. However, it is recommended that the resample step only be carried out when $\boldsymbol{p}^T\boldsymbol{p}$ is above a particular threshold (e.g. Chopin & Papaspiliopoulos, 2020). We provide the justification for this aspect of the approach in Section S.1.1.3 of the supplement.

The *move* step is based on Markov chain theory, and involves use of a transitional kernel with an invariant distribution. Moving the particles helps ensure that the ever-changing posterior density functions are well-approximated by the current particles and their probability vector as new data are observed. For the purposes of online semiparametric regression, the move step simply corresponds to draws from the current full conditional distributions. For illustration, consider the logistic additive model given by (4) and with directed acyclic graph representation in Figure 1. Then the move step is as follows (with $\boldsymbol{C}_{\text{curr}}$ defined below):

For $m = 1, \ldots, M$:

$$
\begin{aligned}
[\boldsymbol{\beta}^T\ \boldsymbol{u}_1^T\ \boldsymbol{u}_2^T]_m^T \longleftarrow\ & \text{draw from the distribution having density function proportional to} \\
& \exp\Big[\boldsymbol{y}_{\text{curr}}^T \boldsymbol{C}_{\text{curr}}[\boldsymbol{\beta}^T\ \boldsymbol{u}_1^T\ \boldsymbol{u}_2^T]^T - \mathbf{1}^T \log\left\{\mathbf{1} + \exp\left(\boldsymbol{C}_{\text{curr}}[\boldsymbol{\beta}^T\ \boldsymbol{u}_1^T\ \boldsymbol{u}_2^T]^T\right)\right\} \\
& \quad -\tfrac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}})^T \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}}) - \frac{\|\boldsymbol{u}_1\|^2}{2\sigma_{u1}^2} - \frac{\|\boldsymbol{u}_2\|^2}{2\sigma_{u2}^2}\Big] \\
(a_{u1})_m \longleftarrow\ & \text{draw from Inverse-Gamma } \left(1, (\sigma_{u1}^2)_m^{-1} + s_{\sigma^2}^{-1}\right) \\
(\sigma_{u1}^2)_m \longleftarrow\ & \text{draw from Inverse-Gamma } \left(\tfrac{1}{2}(K_1+1), (a_{u1})_m^{-1} + \tfrac{1}{2}\|(\boldsymbol{u}_1)_m\|^2\right) \\
(a_{u2})_m \longleftarrow\ & \text{draw from Inverse-Gamma } \left(1, (\sigma_{u2}^2)_m^{-1} + s_{\sigma^2}^{-1}\right) \\
(\sigma_{u2}^2)_m \longleftarrow\ & \text{draw from Inverse-Gamma } \left(\tfrac{1}{2}(K_2+1), (a_{u2})_m^{-1} + \tfrac{1}{2}\|(\boldsymbol{u}_2)_m\|^2\right)
\end{aligned}
$$

With the exception of $[\boldsymbol{\beta}\ \boldsymbol{u}_1\ \boldsymbol{u}_2]$, the particle updates are straightforward. For the $[\boldsymbol{\beta}\ \boldsymbol{u}_1\ \boldsymbol{u}_2]_m$ updates, the matrix $\boldsymbol{C}_{\text{curr}}$ is the current design matrix containing the x_{1i} and x_{2i} predictor data as well as the spline bases evaluated at these predictors. The required draw for updating $[\boldsymbol{\beta}\ \boldsymbol{u}_1\ \boldsymbol{u}_2]_m$ does not involve a standard distribution, and remedies such as Metropolis-Hastings sampling or slice sampling (e.g. Gelman *et al.*, 2014; Chapters 11–12) are required to achieve this part of the move step. Algorithm 7 treats a more general version of (4) and calls upon the random walk proposal version of Metropolis-Hastings sampling for the $[\boldsymbol{\beta}\ \boldsymbol{u}_1\ \boldsymbol{u}_2]_m$ particle moves.

S.1.1.3 Justification of the $\boldsymbol{p}^T\boldsymbol{p}$ Threshold

We now explain the justification for checking that the sum of squares of the sequential Monte Carlo probability vector $\boldsymbol{p}$ is above a particular threshold and, if it is, applying the SYSTEMATICSAMPLE algorithm to the particles.

As in Section 2.2, let θ be a generic model parameter of interest and $\boldsymbol{y}$ denote the current observed data. Then the atoms of the current set of particles

$$\theta_1, \ldots, \theta_M$$

may be thought of as being a weighted sample from the $\theta|\boldsymbol{y}$ distribution with weights stored in $\boldsymbol{p}$. The current approximation to the posterior mean of θ is

$$\sum_{m=1}^{M} \boldsymbol{p}_m \theta_m \quad \text{which has conditional variance} \quad \text{Var}\left(\sum_{m=1}^{M} \boldsymbol{p}_m \theta_m \Big| \boldsymbol{y}\right) = \boldsymbol{p}^T\boldsymbol{p}\,\text{Var}(\theta|\boldsymbol{y}).$$

The extremal situations are

$$\text{Var}\left(\sum_{m=1}^{M} \boldsymbol{p}_m \theta_m \Big| \boldsymbol{y}\right) = \begin{cases} \frac{1}{M}\,\text{Var}(\theta|\boldsymbol{y}) & \text{in the discrete Uniform case} \\ \text{Var}(\theta|\boldsymbol{y}) & \text{in the degenerate case,} \end{cases}$$

where the former case is that where $\boldsymbol{p}_m = 1/M$ for all $1 \leq m \leq M$ and the latter case is that where all of the probability mass is on a single particle. The threshold

$$\boldsymbol{p}\boldsymbol{p}^T > \frac{2}{M} = \text{twice the minimum possible value of the discrete Uniform case} \tag{S.4}$$

serves as a reasonable default for avoiding degeneracy.

Lastly, we relate (S.4) to the *effective sample size* notion from the importance sampling and sequential Monte Carlo literatures (e.g. Chopin & Papaspiliopoulos, 2020; Section 8.6). Trivially,

$$\boldsymbol{p}\boldsymbol{p}^T > \frac{2}{M} \quad \text{is equivalent to} \quad \text{ESS}(\boldsymbol{p}) < \frac{M}{2}$$

where

$$\text{ESS}(\boldsymbol{p}) \equiv 1/\boldsymbol{p}^T\boldsymbol{p} \tag{S.5}$$

is the most common effective sample size measure. Martino *et al.* (2017), for example, provides some alternatives to (S.5).

S.2 Algorithm Justifications

The essence of online semiparametric regression via sequential Monte Carlo is provided by Algorithms 3–7, each of which depend on the Algorithm 1 for the SYSTEMATICSAMPLE steps. We now provide full justifications for them. For completeness, we also justify Algorithm 2, which uses batch processing.

S.2.1 Justification of Algorithm 1

Consider a d-variate discrete distribution with atoms

$$\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M \in \mathbb{R}^d \text{ and probability vector } \boldsymbol{p}\ (M \times 1). \tag{S.6}$$

Algorithm 1 is concerned with obtaining a sample of size M from $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ that is well-behaved in terms of combating degeneracy in sequential Monte Carlo schemes. Systematic resampling is recommended and widely used within the sequential Monte Carlo literature (e.g. Chopin & Papaspiliopoulos, 2020; Chapter 9). It involves the following steps:

1. Let u be a draw from the Uniform$(0, 1)$ distribution.
2. Let $\boldsymbol{\iota}$ be the $(u, u+1, \ldots, u+M-1)/M$ quantiles of the discrete distribution with atoms $(1, \ldots, M)$ and probability vector $\boldsymbol{p}$. This step involves the quantile function definition given at (2).
3. The sample from $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ of size M, with replacement, corresponds the subscripts in $\boldsymbol{\iota}$.

Given $\boldsymbol{p}$ and U, the $\boldsymbol{\iota}$ vector can be obtained in $O(M)$ steps using e.g. the cumulative probabilities-based algorithm given in Table 2 of Li *et al.* (2015). Such an approach is used in Algorithm 1.

S.2.2 Justification of Algorithm 2

Algorithm 2 is a standard Gibbs sampling scheme and involves successive draws from the full conditional distributions of $\boldsymbol{\beta}$, a and σ^2.

Straightforward steps show that full conditional distribution of $\boldsymbol{\beta}$ is

$$N(\boldsymbol{\Omega}^{-1}\boldsymbol{\omega}, \boldsymbol{\Omega}^{-1}) \quad \text{where} \quad \boldsymbol{\Omega} \equiv \frac{\boldsymbol{X}^T\boldsymbol{X}}{\sigma^2} + \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1} \quad \text{and} \quad \boldsymbol{\omega} \equiv \frac{\boldsymbol{X}^T\boldsymbol{y}}{\sigma^2} + \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}}.$$

Therefore, if $\boldsymbol{z} \sim N(\boldsymbol{0}, \boldsymbol{I}_p)$ then

$$\boldsymbol{\Omega}^{-1/2}\boldsymbol{z} + \boldsymbol{\Omega}^{-1}\boldsymbol{\omega} \tag{S.7}$$

is a draw from the full conditional distribution of $\boldsymbol{\beta}$. If $\boldsymbol{\Omega} = \boldsymbol{U}_{\boldsymbol{\Omega}}\text{diag}(\boldsymbol{d}_{\boldsymbol{\Omega}})\boldsymbol{U}_{\boldsymbol{\Omega}}^T$ is the spectral decomposition of $\boldsymbol{\Omega}$ then, noting that $\boldsymbol{U}_{\boldsymbol{\Omega}}^{-1} = \boldsymbol{U}_{\boldsymbol{\Omega}}^T$, it is easy to show that (S.7) is equal to

$$\boldsymbol{U}_{\boldsymbol{\Omega}}\left(\frac{\boldsymbol{U}_{\boldsymbol{\Omega}}^T\boldsymbol{z}}{\sqrt{\boldsymbol{d}_{\boldsymbol{\Omega}}}} + \frac{\boldsymbol{U}_{\boldsymbol{\Omega}}^T\boldsymbol{\omega}}{\boldsymbol{d}_{\boldsymbol{\Omega}}}\right)$$

which corresponds to the $\boldsymbol{\beta}^{[g]}$ update in Algorithm 2. This approach has the advantage that, once the spectral decomposition of $\boldsymbol{\Omega}$ has been obtained, no matrix inversion is required for the $\boldsymbol{\beta}$ draw.

The draws for the auxiliary variable a and variance parameter σ^2 involve routine and simple full conditional derivations.

S.2.3 Justification of Algorithm 3

First note that Algorithm 3 is simply the generic sequential Monte Carlo algorithm for online semiparametric regression of Figure 2 applied to the Bayesian multiple linear regression model (7)–(8).

Let $(\boldsymbol{\beta}_{\text{SMC}})_m$ be the mth column of $\boldsymbol{\beta}_{\text{SMC}}$ and $(\boldsymbol{\sigma}^2_{\text{SMC}})_m$ be the mth entry of $\boldsymbol{\sigma}^2_{\text{SMC}}$. The likelihood of y_{new} as a function of $(\boldsymbol{\beta}_{\text{SMC}})_m$ and $(\boldsymbol{\sigma}^2_{\text{SMC}})_m$ is

$$\begin{aligned}\mathfrak{p}\big(y_{\text{new}}\Big|(\boldsymbol{\beta}_{\text{SMC}})_m,(\boldsymbol{\sigma}^2_{\text{SMC}})_m\big) &= \big\{2\pi(\boldsymbol{\sigma}^2_{\text{SMC}})_m\big\}^{-1/2}\exp\left[-\frac{\big\{y_{\text{new}}-(\boldsymbol{\beta}_{\text{SMC}})_m^T\boldsymbol{x}_{\text{new}}\big\}^2}{2(\boldsymbol{\sigma}^2_{\text{SMC}})_m^2}\right]\\ &\propto \exp\Big[\big\{y_{\text{new}}(\boldsymbol{\beta}_{\text{SMC}})_m^T\boldsymbol{x}_{\text{new}}-\tfrac{1}{2}\{(\boldsymbol{\beta}_{\text{SMC}})_m^T\boldsymbol{x}_{\text{new}}\}^2\big\}/(\boldsymbol{\sigma}^2_{\text{SMC}})_m\\ &\qquad -\tfrac{1}{2}\log\big\{(\boldsymbol{\sigma}^2_{\text{SMC}})_m\big\}\Big]\\ &= \exp\left[\Big(y_{\text{new}}\boldsymbol{\eta}_m-\tfrac{1}{2}\boldsymbol{\eta}_m^2\Big)/(\boldsymbol{\sigma}^2_{\text{SMC}})_m-\tfrac{1}{2}\log\big\{(\boldsymbol{\sigma}^2_{\text{SMC}})_m\big\}\right]\end{aligned}$$

where $\boldsymbol{\eta}=\boldsymbol{\beta}_{\text{SMC}}^T\boldsymbol{x}_{\text{new}}$. Therefore, the updates of the $1\times M$ vector $\boldsymbol{\ell}$ having mth entry equal to $\log\big\{\mathfrak{p}\big(y_{\text{new}}\Big|(\boldsymbol{\beta}_{\text{SMC}})_m,(\boldsymbol{\sigma}^2_{\text{SMC}})_m\big)\big\}$ performed by Algorithm 3 correspond to the Gaussian multiple linear regression special case of the first line of (6) and the $\boldsymbol{\ell}$ vector update is justified.

S.2.4 Justification of Algorithm 4

The batch-based tuning and convergence diagnostics of Algorithm 4 are the Monte Carlo analogues of those conveyed by Algorithm 2′ of Luts *et al.* (2014) for the mean field variational Bayes. In the latter article, such tuning and diagnosis was found to be important for online semiparametric regression with convergence from the simple and naïve initializations not always guaranteed. Similar advice applies to the sequential Monte Carlo approach as demonstrated in Section 5.2.

S.2.5 Justification of Algorithm 5

Algorithm 5 arises from applying the generic sequential Monte Carlo algorithm for online semiparametric regression of Figure 2 to the Gaussian response linear mixed model (10)–(12), but with the batch-based tuning and convergence diagnosis adjustment conveyed by Algorithm 4.

The justification for the $\boldsymbol{\ell}$ vector update is analogous to that for Algorithm 3. The difference here is that the coefficients vector includes the $\boldsymbol{u}_{\text{SMC}}$ random effects component. Also, the $\boldsymbol{x}_{\text{new}}$ from Algorithm 3 is replaced by the $\boldsymbol{c}_{\text{new}}$, the new row of the $\boldsymbol{C}$ matrix that accompanies y_{new}.

The move steps in the second half of the cycle loop of Algorithm 5 correspond to draws from the current full conditional distributions of, in order, the parameters

$$[\boldsymbol{\beta}^T\ \boldsymbol{u}^T]^T,\ a_\varepsilon,\ \sigma^2_\varepsilon,\ a_{u1},\ \sigma^2_{u1},\ldots,a_{uR},\ \sigma^2_{uR}.$$

The derivations of the Gibbsian full conditional distributions involve routine probability calculus steps.

The steps involving the i_{stt}, i_{end} and $\boldsymbol{\omega}$ quantities warrant some explanations. Note that the conditional distribution of the $\boldsymbol{u}$ vector in (10) can be rewritten as

$$\boldsymbol{u}_r\big|\,\sigma^2_{ur}\overset{\text{ind.}}{\sim}N(\boldsymbol{0},\sigma^2_{ur}\,\boldsymbol{I}_{K_r}),\quad 1\le r\le R,$$

where

$$[\boldsymbol{u}_1^T\cdots\boldsymbol{u}_R^T]^T\quad\text{is the partition of }\boldsymbol{u}\text{ such that each }\boldsymbol{u}_r\text{ is }K_r\times 1.$$

For each $1 \leq r \leq R$, the full conditional distribution of σ^2_{ur} is

$$\text{Inverse-Gamma}\big(\tfrac{1}{2}(K_r+1), a_{ur}^{-1} + \tfrac{1}{2}\|\boldsymbol{u}_r\|^2\big) \tag{S.8}$$

The i_{stt} and i_{end} updates ensure that the correct row-wise sub-blocks of $\boldsymbol{u}_{\text{SMC}}$ are used to extract the particles that correspond to the $\boldsymbol{u}_r$ sub-vectors of $\boldsymbol{u}$ ($1 \leq r \leq R$). For each $1 \leq m \leq M$ and $1 \leq r \leq R$, these particles are stored in the temporary vector $\boldsymbol{\omega}$ and then used for the move step corresponding to the (S.8) full conditional distribution.

S.2.6 Justification of Algorithm 6

Algorithm 6 arises from applying the generic sequential Monte Carlo algorithm for online semiparametric regression of Figure 2 to the generalized linear mixed model (18), but with the batch-based tuning and convergence diagnosis adjustment conveyed by Algorithm 4.

If $(\boldsymbol{\beta}_{\text{SMC}})_m$ denotes the mth column of $\boldsymbol{\beta}_{\text{SMC}}$ then the likelihood of y_{new} as a function of $(\boldsymbol{\beta}_{\text{SMC}})_m$ is

$$\mathfrak{p}\big(y_{\text{new}}\big|(\boldsymbol{\beta}_{\text{SMC}})_m\big) = \exp\big\{y_{\text{new}}(\boldsymbol{\beta}_{\text{SMC}})_m^T\boldsymbol{x}_{\text{new}} - b\big((\boldsymbol{\beta}_{\text{SMC}})_m^T\boldsymbol{x}_{\text{new}}\big) + c(y_{\text{new}})\big\} \propto \exp\Big\{y_{\text{new}}\boldsymbol{\eta}_m - b(\boldsymbol{\eta}_m)\Big\}$$

where $\boldsymbol{\eta} = \boldsymbol{\beta}_{\text{SMC}}^T\boldsymbol{x}_{\text{new}}$. Hence, the updates to the vector $1 \times M$ vector $\boldsymbol{\ell}$ having mth entry equal to $\log\big\{\mathfrak{p}\big(y_{\text{new}}\big|(\boldsymbol{\beta}_{\text{SMC}})_m\big)\big\}$ performed by Algorithm 6 correspond to the generalized linear model special case of the first line of (6). This justifies the $\boldsymbol{\ell}$ vector updates.

The move step for the $\boldsymbol{\beta}_{\text{SMC}}$ particles requires draws from the current full conditional distribution of $\boldsymbol{\beta}$, which has density function

$$\mathfrak{p}(\boldsymbol{\beta}|\boldsymbol{y}_{\text{curr}}) \propto \exp\Big\{\boldsymbol{y}_{\text{curr}}^T\boldsymbol{X}_{\text{curr}}\boldsymbol{\beta} - \mathbf{1}^T b(\boldsymbol{X}_{\text{curr}}\boldsymbol{\beta}) - \tfrac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_{\boldsymbol{\beta}})^T\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_{\boldsymbol{\beta}})\Big\}. \tag{S.9}$$

where $\boldsymbol{y}_{\text{curr}}$ and $\boldsymbol{X}_{\text{curr}}$ are, respectively, the current response vector and design matrix. Since (S.9) is not a standard distribution from which draws can be easily made, the particle moves require a non-Gibbsian strategy. Algorithm 6 uses a random walk Metropolis-Hastings approach. The random walk step for possibly updating the mth column of $\boldsymbol{\beta}_{\text{SMC}}$ is

$$\boldsymbol{\beta}_{\text{RW}} \longleftarrow \text{draw from } N\left((\boldsymbol{\beta}_{\text{SMC}})_m, \frac{v^2}{n}\boldsymbol{I}\right) \quad \text{for some } v > 0.$$

The v^2/n factor helps ensure that the posterior variances of the $\boldsymbol{\beta}_{\text{SMC}}$ entries have the familiar $O_P(1/n)$ asymptotic behaviour. Then compute and assign the Metropolis-Hastings ratio

$$\alpha \longleftarrow \frac{\mathfrak{p}(\boldsymbol{\beta}_{\text{RW}}|\boldsymbol{y}_{\text{curr}})}{\mathfrak{p}((\boldsymbol{\beta}_{\text{SMC}})_m|\boldsymbol{y}_{\text{curr}})} \quad \text{as well as} \quad u \longleftarrow \text{draw from Uniform}(0,1).$$

If $\alpha \geq u$ then the assignment $(\boldsymbol{\beta}_{\text{SMC}})_m \longleftarrow \boldsymbol{\beta}_{\text{RW}}$ should be made. Otherwise the mth column of $\boldsymbol{\beta}_{\text{SMC}}$ is not updated. These steps provide the required full conditional draw according to Metropolis-Hastings theory (e.g. Gelman *et al.*, 2014; Chapters 11–12) The $\boldsymbol{\beta}_{\text{SMC}}$ updates in Algorithm 6 are an operationalization of this random walk Metropolis-Hastings approach with the calculations done on the logarithmic scale, which helps avoid underflow/overflow problems.

S.2.7 Justification of Algorithm 7

Algorithm 7 can be justified using arguments already given in the justifications of Algorithms 5 and 6. The $[\boldsymbol{\beta}_{\text{SMC}}^T\ \boldsymbol{u}_{\text{SMC}}^T]^T$ updates of Algorithm 7 are analogous to the $\boldsymbol{\beta}_{\text{SMC}}$ updates of Algorithm 6. The $\boldsymbol{\sigma}^2_{u\text{SMC}}$ updates are identical to those of Algorithm 5.

S.2.8 Frequency Polygonal Visualization of Posterior Distributions

As in Section 2.2 we θ denote a generic univariate parameter in a statistical model that takes values over a continuum and let $\boldsymbol{y}_{\text{curr}}$ denote the currently observed data. The most intuitive way to visualize the posterior distribution of θ is via a plot of the posterior density function $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$. In this subsection we describe an effective approach to visualizing and comparing batch Markov chain Monte Carlo and online sequential Monte Carlo approximations of $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$ based on the classical *frequency polygon* estimator (e.g. Scott, 1985).

In the case of batch Markov chain Monte Carlo, visualization of $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$ typically involves applying a probability density estimation technique, such as kernel density estimation, to the kept sample $\theta^{[1]}, \ldots, \theta^{[N_{\text{kept}}]}$. An alternative approach, which is more amenable to comparison with online sequential Monte Carlo approximations of $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$, is to obtain a histogram from the $\theta^{[1]}, \ldots, \theta^{[N_{\text{kept}}]}$ values and then form the corresponding frequency polygon by connecting with straight lines the mid-bin heights of the histogram bars.

For the online sequential Monte Carlo situation, $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$ is approximated by a probability mass function with atom vector $\boldsymbol{a}_\theta$ and probability vector $\boldsymbol{p}_\theta$. If the number of atoms is high, as is the case for online sequential Monte Carlo, then visualization a probability mass function is challenging. However, the frequency polygonal idea has a natural extension to the discrete case, which lends itself to simple display of an online sequential Monte Carlo approximation $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$ and comparison with the batch Markov chain Monte Carlo counterpart.

Let $\boldsymbol{a}$ be a generic vector of atoms in $\mathbb{R}$ and $\boldsymbol{p}$ be the corresponding probability vector. Suppose that $\mathbb{R}$ is partitioned into equi-sized bins with width h. For each $x \in \mathbb{R}$ define the following function of x:

$$\frac{\text{sum of entries of } \boldsymbol{p} \text{ corresponding to the entries of } \boldsymbol{a} \text{ that are in the bin containing } x}{h}. \quad \text{(S.10)}$$

The function of x defined by (S.10) is piecewise constant over the bins and, therefore, "histogram-like". The frequency polygon representation of the probability mass function with atoms $\boldsymbol{a}$ and probabilities $\boldsymbol{p}$ is formed by connecting with straight lines the mid-bin heights of the function defined by (S.10). Frequency polygonal representation of the online sequential Monte Carlo approximations of $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$ are obtained by applying the principle described in this paragraph to the current $\boldsymbol{a}_\theta$ and $\boldsymbol{p}_\theta$ vectors.

Next we address the practical problem of frequency polygon bin width choice, starting with that for estimation of $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$ based on a batch Monte Carlo Markov chain sample of size N_{kept}. From (2.4) of Scott (1985), the leading term behaviour of the mean integrated squared error optimal bin width, which we denote by h^*_{FP}, is given by

$$h^*_{\text{FP}} = \left[\frac{480}{49 \int_{-\infty}^{\infty} \{\mathfrak{p}''(\theta|\boldsymbol{y}_{\text{curr}})\}^2 \, d\theta} \right]^{1/5} N_{\text{kept}}^{-1/5} \{1 + o(1)\}.$$

The only unknown in the leading term of h^* is the integrated squared second derivative of $\mathfrak{p}(\theta|\boldsymbol{y}_{\text{curr}})$, which is also the only unknown for the optimal bandwidth for kernel density estimation of the same density function – see, for example, (2.13) of Wand & Jones (1995). In the special case of kernel density estimation with the Gaussian kernel $K(x) = (2\pi)^{-1/2} \exp(-x^2/2)$ we have $h^*_{\text{FP}} = \left(1280\sqrt{\pi}/49\right)^{1/5} h^*_{\text{KDE}}\{1 + o(1)\}$ where h_{KDE} is the mean integrated squared error optimal bandwidth for kernel density estimation based on the Gaussian kernel. This leads to the practical frequency polygon bin width rule

$$\widehat{h}_{\text{FP}} = \left(\frac{1280\sqrt{\pi}}{49} \right)^{1/5} \widehat{h}_{\text{KDE}} \quad \text{(S.11)}$$

where $\widehat{h}_{\text{KDE}}$ is a consistent estimator of h^*_{KDE}. This can be obtained from any statistical software package that supports automatic kernel density estimation. For example, in the R language the following code leads to a practical and consistent frequency polygon estimated optimal bin width value based on the Markov chain Monte Carlo sample stored in the array `theta`:

```
library(KernSmooth) ; hHatFP <- (1280*sqrt(pi)/49)^(1/5)*dpik(theta)
```

Here `dpik()` is a function for automatic bandwidth selection within the R package KernSmooth (Wand & Ripley, 2023), and involves consistent estimation of $\int_{-\infty}^{\infty}\{\mathfrak{p}''(\theta|\boldsymbol{y}_{\text{curr}})\}^2\,d\theta$.

The choice of bin width for frequency polygonal representation of the online sequential Monte Carlo probability mass functions remains an open problem. In the comparisons with batch Markov chain Monte Carlo in Figure 3 and corresponding movie in the supplemental material we use the same bin widths for each frequency polygon.

References

Li, T., Bolić, M. & Djurić, P.M. (2015). Resampling methods for particle filtering. *IEEE Signal Processing Magazine*, **32(3)**, 70–86.

Martino, L., Elvira, V. & Louzada, F. (2017). Effective sample size for importance sampling based on discrepancy measures. *Signal Processing*, **131**, 386–401.

Scott, D.W. (1985). Frequency polygons: theory and application. *Journal of the American Statistical Association*, **80**, 348–354.

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*. London: Chapman and Hall.

Wand, M.P. and Ripley, B.D. (2023). KernSmooth 2.23. Functions for kernel smoothing corresponding to the book: Wand, M.P. and Jones, M.C. (1995) "Kernel Smoothing". R package. `https://CRAN.R-project.org/package=KernSmooth`