Australian & New Zealand Journal of Statistics

Aust. N. Z. J. Stat. (2025) doi: 10.1111/anzs.12440

Online semiparametric regression via sequential Monte Carlo

Marianne Menictas[1], Chris J. Oates[2] and Matt P. Wand[3*]

Grubhub Inc., Newcastle University and University of Technology Sydney

Summary

We develop and describe online algorithms for performing online semiparametric regression analyses. Earlier work on this topic is by Luts, Broderick and Wand (2014), *Journal of Computational and Graphical Statistics*, **23**, 589–615, where online mean-field variational Bayes (MFVB) 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 MFVB 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.

Key words: generalised additive models; generalised linear mixed models; real-time algorithms; penalised splines.

1. Introduction

Online semiparametric regression is concerned with rapid online fitting of flexible regression models, such as generalised additive models, and continuously updated inference as data stream in. Luts, Broderick and Wand (2014; LBW hereafter) 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 penalised splines and mean-field variational Bayes (MFVB). The setting and motivations are identical to those of Luts, Broderick & Wand (2014), and background material in the earlier article on the essence of online semiparametric regression also applies here. The crux of this article is the provision of sequential Monte Carlo (SMC) alternatives to the online MFVB approach of Luts, Broderick & Wand (2014).

Online fitting of statistical models for sequentially arriving data has a very long history and large literature. The essential goal is obtaining fits and corresponding inference with

*Author to whom correspondence should be addressed.

[1]Grubhub Inc., 111 West Washington Street, Chicago, Illinois, 60602-2783, USA.

[2]School of Mathematics, Statistics and Physics, Newcastle University, Newcastle upon Tyne, NE1 7RU, UK.

[3]School of Mathematical and Physical Sciences, University of Technology Sydney, P.O. Box 123, Broadway, NSW 2007, Australia. e-mail: matt.wand@uts.edu.au

Acknowledgments. 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.

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) achieved this via SMC. As mentioned earlier, Luts, Broderick & Wand (2014) achieved it via MFVB. The essence of the present article is SMC methodology for online semiparametric regression according to the mixed-model-based splines approach advocated in Ruppert, Wand & Carroll (2003) monograph. This approach has the attraction of requiring only the mixed-model extension of ordinary linear models to achieve nonparametric and semiparametric regression fitting and inference.

The algorithms of Luts, Broderick & Wand (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, Broderick & Wand (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 SMC-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, Broderick & Wand (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 than, but not as fast as, the approach used in Luts, Broderick & Wand (2014). Depending on the speed requirements of the application and volume of data requiring storage, the new SMC 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.

SMC methodology of the type used here originates with Kong, Liu & Wong (1994). Liu & 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 SMC was devised by Del Moral,

Doucet & Jasra (2006). A comprehensive and contemporary overview of SMC is provided by Chopin & Papaspiliopoulos (2020).

Section 2 lays down some preliminary infrastructure that is intrinsic to the SMC 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 that demonstrate the good inferential accuracy of online SMC and contrast it with the patchy performance of online MFVB. Some concluding remarks are made in Section 6.

2. Preliminary infrastructure

Online semiparametric regression via SMC depends on some fundamental concepts and results, which we lay out in this section. Throughout this section, $\mathbf{1}(\mathcal{P})$ 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 2/7, 4/7 and 1/7, respectively. Its *probability mass function* $\mathfrak{p}$ is

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

We say that $\mathfrak{p}$ has *atoms* $\boldsymbol{a} = [5 \quad 11 \quad 13]^T$ and *probabilities* $\boldsymbol{p} = [2/7 \quad 4/7 \quad 1/7]^T$. The corresponding *cumulative distribution function* is, for $x \in \mathbb{R}$

$$F(x; \boldsymbol{a}, \boldsymbol{p}) = (2/7)\mathbf{1}(x \leq 5) + (4/7)\mathbf{1}(x \leq 11) + (1/7)\mathbf{1}(x \leq 13),$$

and *quantile function* is, for $0 \leq q \leq 1$

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

The concepts illustrated here are, of course, very basic and straightforwardly extendable 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 characterised by the probability density function $\mathfrak{p}\left(\theta \,|\, \boldsymbol{y}_{\text{curr}}\right)$. An intrinsic feature of online semiparametric regression via SMC is the sequential approximation of $\mathfrak{p}\left(\theta \,|\, \boldsymbol{y}_{\text{curr}}\right)$ by *probability mass functions* as new observations arrive. To repeat, even though $\mathfrak{p}\left(\theta \,|\, \boldsymbol{y}_{\text{curr}}\right)$ 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 $\boldsymbol{y}_{\text{curr}} \leftarrow \left(\boldsymbol{y}_{\text{curr}}, y_{\text{new}}\right)$. The current posterior density function of

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 $m = 1, \ldots M$:

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

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

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

θ, $\mathfrak{p}\left(\theta \mid \boldsymbol{y}_{\text{curr}}\right)$, 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 $\left(\boldsymbol{p}_\theta\right)^\top \boldsymbol{a}_\theta$ and, for example, a current approximate 95% credible interval for θ is

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

where $Q\left(\cdot; \boldsymbol{a}_\theta, \boldsymbol{p}_\theta\right)$ 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 SMC procedures is 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 SMC 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.

An example of the last step of SystematicResample is as follows: if $d = 3$, $M = 5$ and $\iota = (3, 3, 5, 2, 2)$, 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 Supporting Information.

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

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

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 $\sigma \sim \text{Half-Cauchy}(s)$ means that σ has the density function

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

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

$$\sigma^2 \mid a \sim \text{Inverse-Gamma}\left(\frac{1}{2}, 1/a\right), \quad a \sim \text{Inverse-Gamma}\left(\frac{1}{2}, 1/s^2\right). \tag{3}$$

Representation (3) has the attraction of leading to draws from standard distributions in our SMC 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 a column vector, then $\|\boldsymbol{a}\| = \sqrt{\boldsymbol{a}^\top \boldsymbol{a}}$ 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 $\exp\left([7 \;\; 4 \;\; 6]^\top\right) = [\exp(7) \;\; \exp(4) \;\; \exp(6)]^\top$. 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 nonlinear effects are handled using suitable basis functions and penalisation (e.g., Ruppert, Wand & Carroll 2003). Special cases include nonparametric regression, generalised additive models, varying-coefficient models and generalised additive mixed models. The version of semiparametric regression that we use here is Bayesian models for which the nonlinear effects correspond to mixed-model-based penalised splines. As an illustrative example, consider the regression-type dataset with responses $y_i \in \{0, 1\}$ and bivariate continuous predictors (x_{1i}, x_{2i}), $1 \le i \le n$. Then a *logistic additive model* is

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

where $\overset{\text{ind.}}{\sim}$ stands for 'independently distributed as' and $\text{expit}(x) = 1/\left(1 + e^{-x}\right)$. The functions $\{z_{1k}(\cdot) : 1 \le k \le K_1\}$ and $\{z_{2k}(\cdot) : 1 \le k \le K_2\}$ are suitable spline bases (see e.g.,

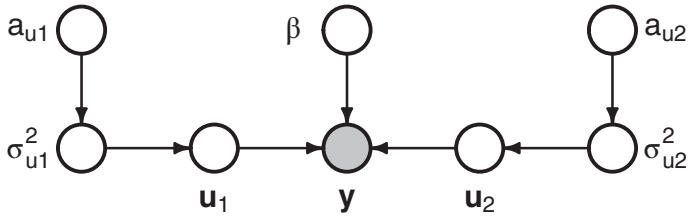

Figure 1. Directed acyclic graph corresponding to the Bayesian logistic additive model (4). The shading indicates that the $\boldsymbol{y}$ node is observed.

Wand & Ormerod 2008) for the x_1 and x_2 nonlinear 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 requires inference.

Model (4) and its Figure 1 representation exemplifies the approach to Bayesian semiparametric regression used here, with spline basis function penalisation 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, Ruppert & Wand 2018). Our concern here, though, is online fitting of (4) as data stream in. Algorithm 5 of Luts, Broderick & Wand (2014) provides a solution to this problem using online MFVB. However, this approach is susceptible to poor Bayesian inferential accuracy. Therefore, the SMC alternative is being pursued here.

Typical Bayesian semiparametric regression models have between 10s and 100s 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 let $\boldsymbol{\theta}$ 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 $M = 1000$.

Algorithm 2 conveys the SMC 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 Supporting Information. Most of the steps in Algorithm 2 involve simple calculations. The possible exception is the step involving drawing independent samples from the 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 depend only 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 generalised response situation, with model (4) as an example, is more challenging because of the current full conditional distributions having nonstandard 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 Supporting Information, the general form of the probability vector updates when a new response y_{new} and its corresponding predictor data vector arrive is

Algorithm 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.

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.

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

Straightforward algebraic arguments then lead to the updating steps:

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

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

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

where

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

and $(x_{1\mathrm{new}}, x_{2\mathrm{new}})$ 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 SMC 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 an $n \times p$ design matrix, and consider the Bayesian regression model

$$\boldsymbol{y} \mid \boldsymbol{\beta}, \sigma^2 \sim \mathrm{N}\left(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}\right), \quad \boldsymbol{\beta} \sim \mathrm{N}\left(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \Sigma_{\boldsymbol{\beta}}\right), \quad \sigma \sim \text{Half-Cauchy}\left(s_{\sigma^2}\right) \tag{7}$$

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

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

is replaced by the auxiliary variable representation

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

Batch-fitting of (7) via Markov Chain Monte Carlo (MCMC) is very well established (e.g., Gelman *et al.* 2014, Chapters 11–12) and it is listed in Algorithm 3. It relies on the result (e.g., Tierney 1994) that, after convergence can be assumed following the 'burn-in' phase of length N_{burn},

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

Note that Algorithm 3 uses the spectral decomposition of the matrix denoted by Ω to efficiently obtain draws from the full conditional distribution of the $\boldsymbol{\beta}$ vector. The justifications for this and other aspects of Algorithm 3 are given in Section S.2.2 of Supporting Information.

Algorithm 4 is the online counterpart of Algorithm 3, based on the general approach of Algorithm 2. Algorithm 4 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 do not have to be stored or used again after the sufficient statistics have been updated. In this sense, Algorithm 4 achieves purely online fitting and inference

Algorithm 3. 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: $\boldsymbol{\mu}_{\boldsymbol{\beta}}$ $(p \times 1)$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ $(p \times p)$ 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 $g = 1, \ldots, N_{\text{burn}} + N_{\text{kept}}$:

$$\boldsymbol{\Omega} \longleftarrow \frac{\mathbf{XTX}}{(\sigma^2)^{[g-1]}} + \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}$$

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 $\text{N}(0, 1)$ draws

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

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

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

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)).

according to the definition described in Section 1. Section S.2.3 of Supporting Information provides justifications for the Algorithm 4 steps.

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

Algorithm 5 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:

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

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_ε^2 is the error variance and

Algorithm 4. 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 $\text{N}(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$
$\mathbf{a}_{\text{SMC}} \longleftarrow 1 \times M$ vector containing arandom sample from Inverse-Gamma$(1/2, 1/s_{\varepsilon}^2)$
$\boldsymbol{\sigma}^2_{\text{SMC}} \longleftarrow 1 \times M$ vector containing a random sample from Half-Cauchy(s_{σ^2})
$\ell \longleftarrow \log\big(1/M\big)\mathbf{1}$; $n \longleftarrow 0$; $\mathbf{yTy} \longleftarrow 0$
$\mathbf{XTy} \longleftarrow \mathbf{0}$ $(p \times 1)$; $\mathbf{XTX} \longleftarrow \mathbf{0}$ $(p \times p)$.

Cycle:

Read in $y_{\text{new}}(1 \times 1)$ and $\boldsymbol{x}_{\text{new}}(p \times 1)$; $n \longleftarrow n + 1$
$\mathbf{yTy} \longleftarrow \mathbf{yTy} + y^2_{\text{new}}$; $\mathbf{XTy} \longleftarrow \mathbf{XTy} + \boldsymbol{x}_{\text{new}} y_{\text{new}}$
$\mathbf{XTX} \longleftarrow \mathbf{XTX} + \boldsymbol{x}_{\text{new}} \boldsymbol{x}^T_{\text{new}}$; $\boldsymbol{\eta} \longleftarrow \boldsymbol{\beta}^T_{\text{SMC}} \boldsymbol{x}_{\text{new}}$
$\ell \longleftarrow \ell + (y_{\text{new}} \boldsymbol{\eta} - \frac{1}{2}\,\boldsymbol{\eta} \odot \boldsymbol{\eta})/\{(\boldsymbol{\sigma}^2_{\text{SMC}})^T\} - \frac{1}{2}\log\{(\boldsymbol{\sigma}^2_{\text{SMC}})^T\}$
$\boldsymbol{p} \longleftarrow \exp\{\ell - \max(\ell)\}/[\mathbf{1}^T \exp\{\ell - \max(\ell)\}]$
If $\boldsymbol{p}^T\boldsymbol{p} > \tau$ then

$$\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \mathbf{a}_{\text{SMC}} \\ \boldsymbol{\sigma}^2_{\text{SMC}} \end{bmatrix} \longleftarrow \textsc{SystematicResample}\left(\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \mathbf{a}_{\text{SMC}} \\ \boldsymbol{\sigma}^2_{\text{SMC}} \end{bmatrix}, \boldsymbol{p}\right)$$

$\ell \longleftarrow \log\big(1/M\big)\mathbf{1}$

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

$$\boldsymbol{\Omega} \longleftarrow \frac{\mathbf{XTX}}{(\boldsymbol{\sigma}^2_{\text{SMC}})_m} + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{\beta}}$$

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 $\text{N}(0, 1)$ draws

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

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

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

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.

Algorithm 5. Modification of Algorithm 4 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 3 with $N_{\text{kept}} = M$. Use the kept chains $\boldsymbol{\beta}^{[g]}$, $(\sigma^2)^{[g]}$ and $a^{[g]}$, $1 \le g \le M$, to initialise $\boldsymbol{\beta}_{\text{SMC}}$, $\mathbf{a}_{\text{SMC}}$ 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 4 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 andincrease n_{warm}.
   (b) If converging then continue running the online sequential Monte Carlo Algorithm 4 until data no longer available or analysis terminated.

$\sigma^2_{u1}, \ldots, \sigma^2_{ur}$ are variance parameters corresponding to sub-blocks of $\boldsymbol{u}$ of size $K_1, \ldots, K_R$. We set the priors to be

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

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

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

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

As described in Section 2 of Zhao *et al.* (2006), models (10) and (11) cover 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),
- generalised additive models (Example 6),
- semiparametric mixed models (Example 7),
- bivariate smoothing and geoadditive model 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 let P be the number of columns in $\boldsymbol{C}$. Then, each pass of the corresponding online SMC 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 6 for purely online fitting of (10). Its justification is given in Section S.2.5 of Supporting Information.

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

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

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

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

is an unstructured 2×2 covariance matrix. The conjugate prior for Σ is the Inverse-Wishart distribution. Note that

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

provides a covariance matrix extension of $\sigma_u \sim \text{Half-Cauchy}\,(A_u)$. The choice $\nu = 2$ imposes a $\text{Uniform}\,(-1, 1)$ 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 $\text{Inverse-Wishart}\,(a, \boldsymbol{B})$ distribution. Extension to models with larger unstructured covariance matrices is similar.

4. Generalised response models

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

4.1. Generalised linear models

As with the (7) set-up, let $\boldsymbol{X}$ be an $n \times p$ 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 $\mathbf{y}$ have the following one-parameter exponential family probability mass or density function:

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

Algorithm 6. 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, $s_{\sigma^2} > 0$, $s_{ur} > 0$, $1 \le r \le R$.
Perform batch-based tuning runs analogous to those described in Algorithm 5 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)
$\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}$ $(P \times M)$; $\boldsymbol{\sigma}^2_{\varepsilon\text{SMC}}$ $(1 \times M)$; $a_{\varepsilon\text{SMC}}$ $(1 \times M)$; $\boldsymbol{\sigma}^2_{u\text{SMC}}$ $(r \times M)$;
$\mathbf{a}_{u\text{SMC}}$ $(r \times M)$ 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_{new} (1×1) and $\boldsymbol{c}_{\text{new}}$ $(P \times 1)$; $n \leftarrow n + 1$

$$\boldsymbol{\eta} \longleftarrow \begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}^T \boldsymbol{c}_{\text{new}}$$

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

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

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

$$\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \\ \boldsymbol{\sigma}^2_{\varepsilon\text{SMC}} \\ a_{\varepsilon\text{SMC}} \\ \boldsymbol{\sigma}^2_{u\text{SMC}} \\ \mathbf{a}_{u\text{SMC}} \end{bmatrix} \longleftarrow \text{SystematicResample}\left(\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \\ \boldsymbol{\sigma}^2_{\varepsilon\text{SMC}} \\ a_{\varepsilon\text{SMC}} \\ \boldsymbol{\sigma}^2_{u\text{SMC}} \\ \mathbf{a}_{u\text{SMC}} \end{bmatrix}, \boldsymbol{p} \right)$$

$$\ell \longleftarrow \log\big(1/M\big)\mathbf{1}$$

$\mathbf{yTy} \leftarrow \mathbf{yTy} + y^2_{\text{new}}$; $\mathbf{CTy} \leftarrow \mathbf{CTy} + \boldsymbol{c}_{\text{new}}\, y_{\text{new}}$; $\mathbf{CTC} \leftarrow \mathbf{CTC} + \boldsymbol{c}_{\text{new}}\, \boldsymbol{c}^T_{\text{new}}$
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 $\boldsymbol{\Omega} = \boldsymbol{U}_{\boldsymbol{\Omega}}\text{diag}(\boldsymbol{d}_{\boldsymbol{\Omega}})\boldsymbol{U}^T_{\boldsymbol{\Omega}}$ 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
continued on a subsequent page ...

Algorithm 6. continued

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

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

$$(\boldsymbol{\sigma}^2_{\varepsilon\text{SMC}})_m \sim \text{Inverse-Gamma}\Bigg((n+1)/2, (a_{\varepsilon\text{SMC}})_m^{-1} + (1/2)\left\{ \mathbf{yTy} - 2\left((\mathbf{CTy})^T \begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix} \right)_m + \left(\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}^T \mathbf{CTC} \begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix} \right)_{mm} \right\} \Bigg)$$

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

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

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

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

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

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

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

Produce summaries based on the current approximate posterior distributions of $\begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{u} \end{bmatrix}$, σ_ε^2 and $(\sigma_{u1}^2, \ldots, \sigma_{uR}^2)$ equalling theprobability 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.

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

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

Instead, setting

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

corresponds to Poisson regression for count responses.

Online fitting of (14) is provided by Algorithm 7 and justified in Section S.2.6 of Supporting Information. An important difference between Algorithm 7 for generalised linear models and Algorithm 4 for Gaussian response linear models is that purely online fitting is *not* being achieved. Recall that Algorithm 4 is such that the data can be discarded after

Algorithm 7. Online sequential Monte Carlo algorithm for online approximate inference in the generalised 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 5 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 $\boldsymbol{x}_{\text{new}}(p \times 1)$; $n \longleftarrow n + 1$

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

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

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

$$\boldsymbol{\beta}_{\text{SMC}} \longleftarrow \text{SYSTEMATICRESAMPLE}\big(\boldsymbol{\beta}_{\text{SMC}}, \boldsymbol{p}\big) \quad ; \quad \ell \longleftarrow \log\big(1/M\big)\mathbf{1}$$

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

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

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

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

$u \longleftarrow$ 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.

each sufficient statistic update is accomplished. In contrast, Algorithm 7 is such that, every time a new data vector arrives, the full data to date need 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 generalised 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 7 by λ, requires the full data to date.

4.1.1. *Choice of the Metropolis–Hastings random walk scale parameter*

Algorithm 7 involves the following step:

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

for some choice of the scale parameter $\upsilon > 0$. As explained in Section S.2.6 of Supporting Information, (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 & Taylor (2013) were among the first to bring ideas from adaptive MCMC to bear in the context of SMC; they suggested the adaptation of υ based on the expected squared jumping distance, see also Bon, Lee & Drovandi (2021). Maximising the expected squared jumping distance is equivalent to minimising the first-order autocorrelation of the Markov chain, and computation is straightforward. Section 17.2.1 of Chopin & Papaspiliopoulos (2020) recommends the use of an estimate of the sample covariance from the previous step of SMC 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, the 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 7 step:

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

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. Generalised linear mixed models

The class of Bayesian generalised linear mixed models that we consider is

$$\begin{aligned} &\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{u}) = \exp\left\{\boldsymbol{y}^{\top}(\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{Z}\boldsymbol{u}) - 1^{\top}(\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{Z}\boldsymbol{u}) + 1^{\top}\boldsymbol{c}(\boldsymbol{y})\right\} h(\boldsymbol{y}), \\ &\boldsymbol{u}|\sigma_{u1}^2, \ldots, \sigma_{\mathrm{uR}}^2 \sim \mathrm{N}\left(0, \mathrm{blockdiag}\left(\sigma_{u1}^2 \boldsymbol{I}_{K_1}, \ldots, \sigma_{\mathrm{uR}}^2 \boldsymbol{I}_{K_r}\right)\right), \end{aligned} \tag{18}$$

where $\boldsymbol{\beta}$ and σ_{ur}^2, $1 \le r \le R$, have prior distributions as given by (11). Model (18) has similar utility to (10) for various semiparametric regression scenarios, but for generalised 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 8 describes online fitting of (18) via SMC, with justification provided by Section S.2.7 of Supporting Information. An illustration of Algorithm 8 is given in Section 5.2.

Algorithm 8. Online sequential Monte Carlo algorithm for approximate inference in the generalised 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 5 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)

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

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_{new} (1×1) and $\boldsymbol{c}_{\text{new}}$ $(P \times 1)$; $n \leftarrow n + 1$
$\boldsymbol{\eta} \longleftarrow \begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}^T \boldsymbol{c}_{\text{new}}$; $\ell \longleftarrow \ell + y_{\text{new}}\boldsymbol{\eta} - b(\boldsymbol{\eta})$
$\boldsymbol{p} \longleftarrow \exp\{\ell - \max(\ell)\}/[\mathbf{1}^T \exp\{\ell - \max(\ell)\}]$
If $\boldsymbol{p}^T\boldsymbol{p} > \tau$ then

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

$\ell \longleftarrow \log\big(1/M\big)\mathbf{1}$

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

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

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

$$\begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}_m \longleftarrow m\text{th column of } \begin{bmatrix} \boldsymbol{\beta}_{\text{SMC}} \\ \boldsymbol{u}_{\text{SMC}} \end{bmatrix}$$

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

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

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

continued on a subsequent page ...

Algorithm 8. continued

$i_{\text{stt}} \longleftarrow i_{\text{end}} + 1$
$\lambda \longleftarrow \lambda - \frac{1}{2}\left(\|\boldsymbol{\omega}_{\text{SMC}}\|^2 - \|\boldsymbol{\omega}_{\text{RW}}\|^2\right) \Big/ (\boldsymbol{\sigma}^2_{u\text{SMC}})_{rm}$
$u \longleftarrow$ draw from the Uniform$(0, 1)$ distribution
if $\lambda > \log(u)$ then

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

$i_{\text{stt}} \longleftarrow 1$
For $r = 1, \ldots, R$:

$(\mathbf{a}_{u\text{SMC}})_{rm} \sim \text{Inverse-Gamma}\left(1, (\boldsymbol{\sigma}^2_{u\text{SMC}})^{-1}_{rm} + s^{-1}_{ur}\right)$
$i_{\text{end}} \longleftarrow i_{\text{stt}} + K_r - 1$
$\boldsymbol{\omega} \longleftarrow$ entries i_{stt} to i_{end} of the mth column of $\boldsymbol{u}_{\text{SMC}}$
$i_{\text{stt}} \longleftarrow i_{\text{end}} + 1$
$(\boldsymbol{\sigma}^2_{u\text{SMC}})_{rm} \sim \text{Inverse-Gamma}\left((K_r + 1)/2, (\mathbf{a}_{u\text{SMC}})^{-1}_{rm} + \frac{1}{2}\|\boldsymbol{\omega}\|^2\right)$.

Produce summaries based on the current approximate posterior distributions of $\begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{u} \end{bmatrix}$ 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 4–8 on many simulated and actual datasets. In this section we give some illustrations of the practical performance of the new methodology. The first one includes a comparison with the Luts, Broderick & Wand (2014) variational approach.

5.1. Online logistic regression

Algorithm 5 of Luts, Broderick & Wand (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 MFVB approach to logistic regression is susceptible to poor accuracy. This problem, as well as the online SMC 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

$$x_{\text{new}} \sim \text{Uniform}\,(0, 1), \quad y_{\text{new}} \mid x_{\text{new}} \sim \text{Ber}\left(\text{expit}\left(\beta_{0,\text{true}} + \beta_{1,\text{true}}\, x_{\text{new}}\right)\right) \tag{19}$$

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

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

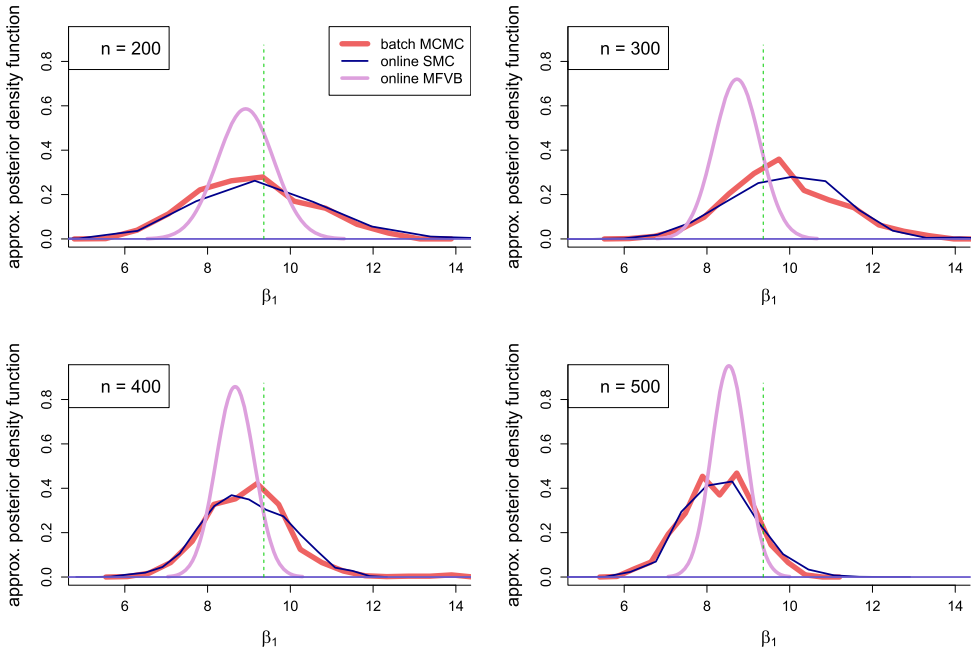

Figure 2. Comparison of the approximations of $\mathfrak{p}\left(\beta_1 \,|\, \boldsymbol{x}_{\text{curr}}, \boldsymbol{y}_{\text{curr}}\right)$ for the online logistic regression example. The batch Markov Chain Monte Carlo (MCMC) approximation is displayed as a frequency polygon density estimate based 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 Supporting Information, and same bin width as the MCMC frequency polygon. The online mean-field variational Bayes (MFVB) approximations, corresponding to Algorithm 5 of Luts, Broderick & Wand (2014), are Normal density functions. The dashed vertical line corresponds to $\beta_{1,\ true} = 9.36$.

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

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

We warmed up both Algorithm 7 of the present paper and Algorithm 5 of Luts, Broderick & Wand (2014) with a sample size of $n_{\ warm} = 100$ and terminated at $n = 500$. As a 'gold standard' we also obtain the batch MCMC fits to each of the $n = 100, 101, \ldots, 500$ datasets using the package rstan (Guo *et al.* 2023) within the R computing environment (R Core Team 2024). A Movie S1 in Supporting Information displays and compares the approximations of $\mathfrak{p}\left(\beta_0 \,|\, \boldsymbol{x}_{\text{curr}}, \boldsymbol{y}_{\text{curr}}\right)$ and $\mathfrak{p}\left(\beta_1 \,|\, \boldsymbol{x}_{\text{curr}}, \boldsymbol{y}_{\text{curr}}\right)$. Figure 3 shows four frames of the movie for $n \in \{200, 300, 400, 500\}$ and the parameter β_1. The approximations to the $\mathfrak{p}\left(\beta_j \,|\, \boldsymbol{x}_{\text{curr}}, \boldsymbol{y}_{\text{curr}}\right)$ based on batch MCMC and online SMC are displayed using frequency polygons, as described in Section S.2.8 of Supporting Information with bin width rule (S.11). The online MFVB approximations are Normal density functions.

Figure 2 and its extended movie form show that online SMC leads to very good approximation of the posterior distributions of β_0 and β_1. In contrast, online MFVB provides overly narrow posterior density function approximations for this example.

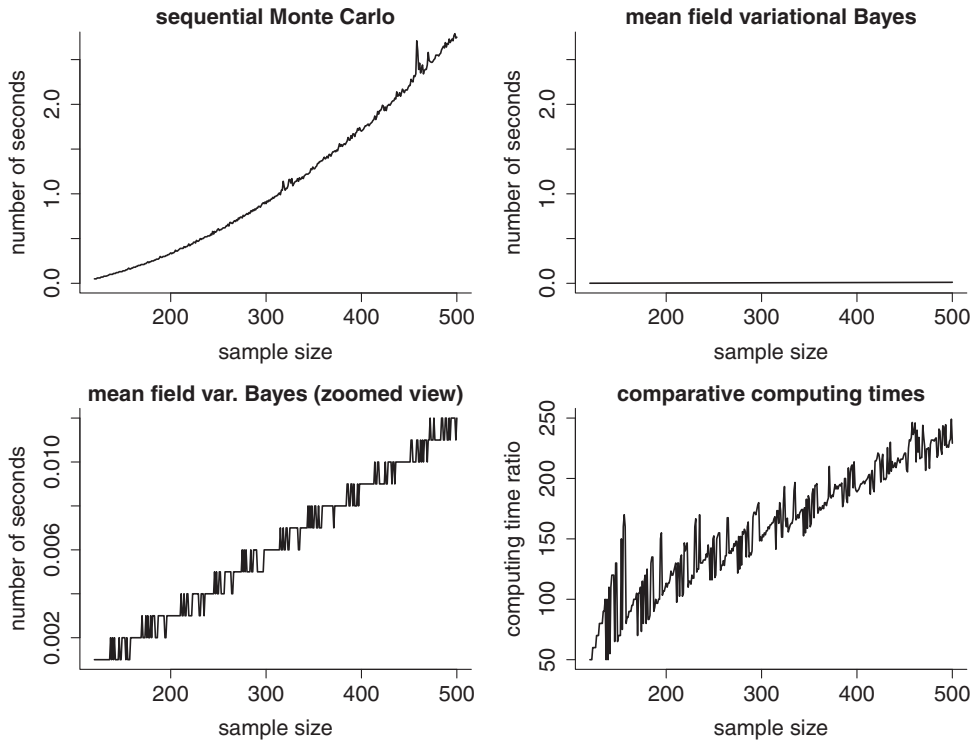

Figure 3. The computing times and their ratios for the Figure 3 example. In each panel, the horizontal axis corresponds to the sample size in the online fitting phase. In the upper panels, the vertical axis corresponds to number of seconds required for online fitting and inference after the warm-up phase; 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.

A price to be paid for SMC's improved accuracy is slowness (higher computing time). Figure 3 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 GHz processor and 16 gigabytes of random access memory. The jaggedness in the Figure 3 curves is due to rounding. It takes SMC about 2.75 s to get from $n = 100$ to $n = 500$, whereas MFVB takes only 0.01 s. The Figure 3 curves show the times and their ratios for getting from $n = 100$ to intermediate sample sizes. The SMC computing times appear to be quadratic in sample size, while the MFVB times are linear. The 'lightning fast' aspect of the online MFVB needs to be traded off against it being prone to inaccurate inference, as Figure 2 demonstrates.

5.2. Online binary response nonparametric regression

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

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

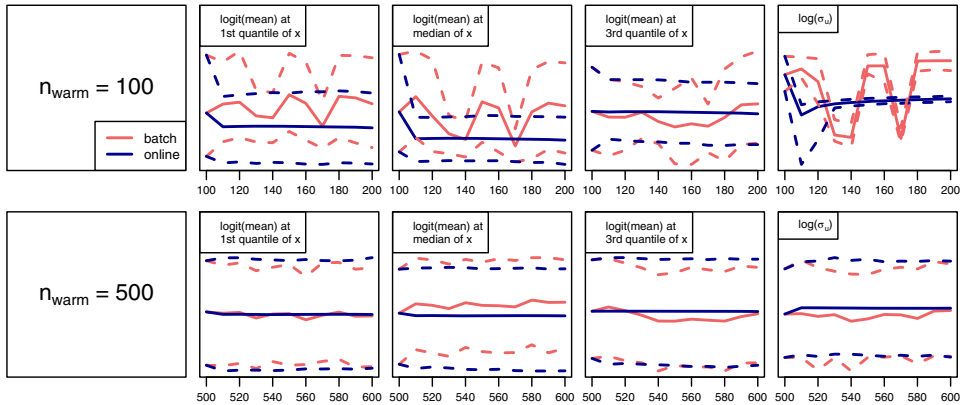

Figure 4. Batch-based convergence diagnostics for the binary response nonparametric regression example. The solid lines track posterior means, while the dashed lines show the limits of the corresponding 95% credible intervals. First row: the horizontal axes correspond to sample sizes between a warm-up 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 $n_{warm} = 500$.

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

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

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

Online estimation and inference concerning f_{true} can be achieved using the $R = 1$ version of Algorithm 8 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 described in Section 4 of Wand & Ormerod (2008). The $\boldsymbol{C}$ matrix appearing in Algorithm 8 is then given by $\boldsymbol{C} = [\boldsymbol{X}\ \boldsymbol{Z}]$. For this example we have $K = 37$.

With the online SMC particles having a dimension of around 40, we found that a substantial batch-based warm-up was required. This aspect is conveyed by Figure 4, which compares the diagnostic plots corresponding to the generalised linear mixed-model extension of Algorithm 5 for $n_{\text{warm}} = 100$ and $n_{\text{warm}} = 500$. It is apparent from Figure 4 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 MFVB approach used by Luts, Broderick & Wand (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

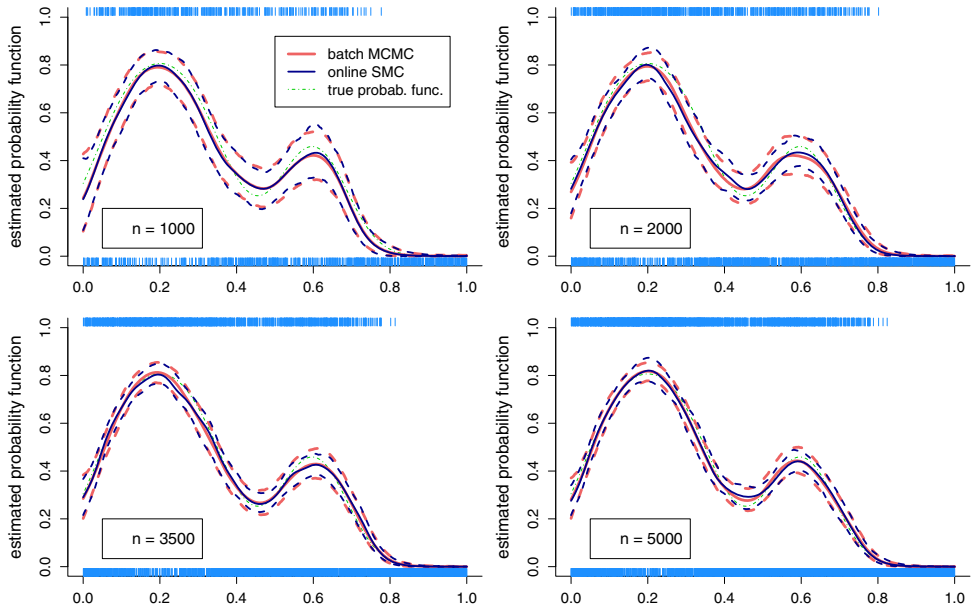

Figure 5. Comparison of online sequential Monte Carlo and batch Markov chain Monte Carlo inference for the probability function f_{true} in the binary response nonparametric regression example for four example sample sizes from the Movie S2 in Supporting Information. The solid curves correspond to the posterior mean, which is targeting the true probability function shown as a dot-dashed curve. The dashed curves correspond to pointwise 95% credible intervals.

values corresponding to lower n_{warm} are more susceptible to divergence away from the correct posterior distributions.

A Movie S2 in Supporting Information of this article displays and compares the online SMC and batch MCMC estimates, and variability, bands of f_{true}. Figure 5 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 6 for online additive model fitting to sequentially arriving data from the data frame `SydneyRealEstate` within the R package HRW (Harezlak, Ruppert & Wand 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 nonlinearly. Given that this is just an illustration, we first ran the full data through the generalised 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 nonlinearly. 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

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

Predictors entering linearly	Predictors entering nonlinearly
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 micrometres	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

and approximate non-informativity was imposed through the use of the hyperparameter choices $\boldsymbol{\mu}_{\boldsymbol{\beta}} = \mathbf{0}$, $\Sigma_{\boldsymbol{\beta}} = 10^{10}\boldsymbol{I}$ 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 6 was warmed up with a batch MCMC 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 $n \in \{1000, 1010, 1020, \ldots, 5000\}$ datasets. A Movie S3 in Supporting Information shows the online additive model fits and compares them with their batch counterparts. Figure 6 shows the $n = 3000$ frame of the movie. In both the movie and Figure 6, 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 Supporting Information. The nonlinear 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 6 demonstrate online Bayesian inference via Algorithm 6 essentially matching the results from successive batch analyses.

6. Concluding remarks

We have demonstrated that SMC provides a viable approach to online, or real-time, semiparametric regression that overcomes the accuracy shortcomings of the MFVB approach. Our algorithms facilitate straightforward implementation in a wide range of semiparametric regression settings. For generalised 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.

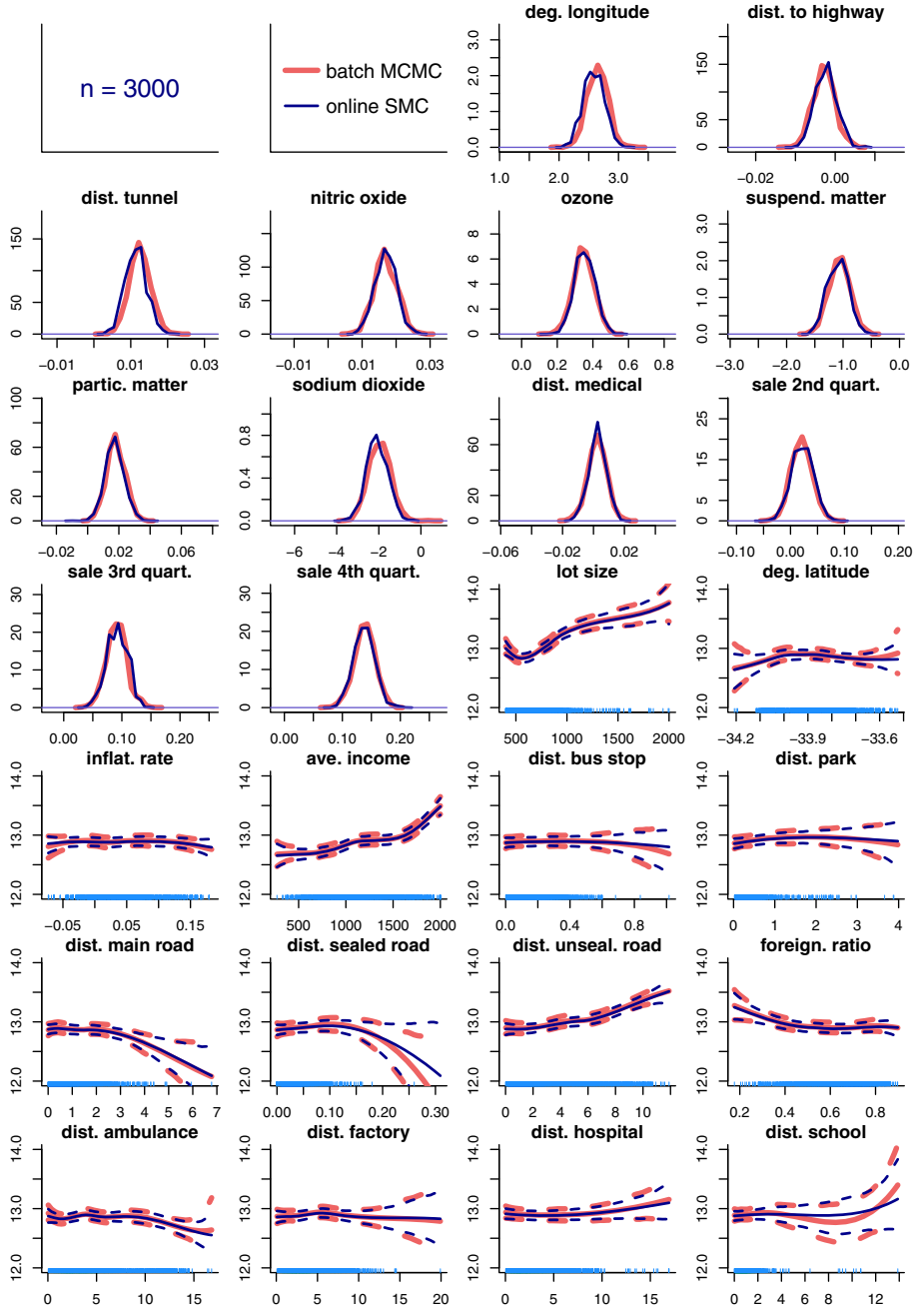

Figure 6. Comparison of online sequential Monte Carlo and batch Markov Chain Monte Carlo inference for the additive model fits to the Sydney real estate data for a sample size of $n = 3000$. The frequency polygon plots correspond to approximate posterior density functions of the coefficients for predictors 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.

Supporting information

Additional supporting information may be found in the online version of this article at http://wileyonlinelibrary.com/journal/anzs.

Movie S1. MOWlogisRegn.mov: Comparison of the online sequential Monte Carlo, the online mean-field variational Bayes and the batch Markov Chain Monte Carlo approaches for the logistic regression example described in Section 5.1. For each sample size from 100 to 500, the movie shows the approximate posterior density functions of the intercept and slope parameters according each of the three approaches. The true values are shown as vertical dashed lines.

Movie S2. MOWbinNPR.mov: Comparison of the online sequential Monte Carlo and the batch Markov Chain Monte Carlo approaches for the binary response nonparametric regression example described in Section 5.2. For each sample size from 500 to 5000, the movie shows the posterior means as solid curves, which are targeting the true probability function shown as a dot-dashed curve. The dashed curves correspond to pointwise 95% credible intervals.

Movie S3. MOWaddModSydReaEst.mov: Comparison of the online sequential Monte Carlo and the batch Markov Chain Monte approaches for the example described in Section 5.3, concerned with online additive model fitting of the Sydney real estate data. The movie shows and compares Bayesian inferential summaries for sample sizes from 1000 to 5000. 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.

Document S1 Supplemental Material.

References

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

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*, 3rd edn. 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). https://CRAN.R-project.org. R package version 1.0.

HE, V.X. & WAND, M.P. (2023). gamselBayes: Bayesian generalized additive model selection. https://CRAN.R-project.org. R package version 2.0.

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*, eds., Grossman, W., Pflug, G.C., & Wertz, W., pp. 191–209. Dordrecht, Netherlands: D. Reidel.

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. https://CRAN.R-project.org. R package version 4-13.

R CORE TEAM (2024). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. 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. https://mc-stan.org.Version 2.30.

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**, 13–37.

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 & 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.

ZHAO, Y., COULL, B.A., STAUDENMAYER, J. & WAND, M.P. (2006). General design Bayesian generalized linear mixed models. *Statistical Science*, **21**, 35–51.