Australian & New Zealand Journal of Statistics

Aust. N. Z. J. Stat. **57**(1), 2015, 119–138 doi: 10.1111/anzs.12105

VARIATIONAL INFERENCE FOR HETEROSCEDASTIC SEMIPARAMETRIC REGRESSION

MARIANNE MENICTAS* AND MATT P. WAND

University of Technology Sydney

Summary

We develop fast mean field variational methodology for Bayesian heteroscedastic semiparametric regression, in which both the mean and variance are smooth, but otherwise arbitrary, functions of the predictors. Our resulting algorithms are purely algebraic, devoid of numerical integration and Monte Carlo sampling. The locality property of mean field variational Bayes implies that the methodology also applies to larger models possessing variance function components. Simulation studies indicate good to excellent accuracy, and considerable time savings compared with Markov chain Monte Carlo. We also provide some illustrations from applications.

Key words: Approximate Bayesian inference; mean field variational Bayes; non-conjugate variational message passing; variance function estimation.

1. Introduction

Data sets that are big in terms of volume and/or velocity are becoming widespread and there is a strong imperative for the development of fast, flexible and extendable methodology for processing such data. Semiparametric regression (e.g. Ruppert, Wand & Carroll 2003, 2009) is an important class of flexible statistical models and methods, but is mainly geared towards moderate sample sizes and batch processing. Recently, Luts, Broderick & Wand (2014) developed new nonparametric regression methodology specifically tailored to high volume/velocity situations. The present article extends their general approach to accommodate possible heteroscedasticity. Whilst we focus on univariate and bivariate nonparametric regression and additive models, the modularity of our approach allows easy extension to more complex models. The accommodation of heteroscedasticity is aided by the variational inference technique known as non-conjugate variational message passing (e.g. Knowles & Minka 2011). The particular form of non-conjugate variational message passing that we use involves approximating particular posterior density functions by Multivariate Normal density functions, which is an established practice within the variational approximation literature (e.g. Hinton & van Camp 1993; Barber & Bishop 1997; Raiko *et al.* 2007; Challis & Barber 2013).

In nonparametric regression it is common to ignore heteroscedasticity in the data and invoke the constant variance assumption. Figure 1 contains nonparametric regression examples, based on the R function `smooth.spline()` (R Core Team 2015) with default settings.

*Author to whom correspondence should be addressed.

School of Mathematical Sciences, University of Technology Sydney, Broadway, 2007, Australia.
e-mail: marianne.menictas@uts.edu.au.

Acknowledgements. This research was partially supported by an Australian Postgraduate Award and Australian Research Council Discovery Project DP110100061. The authors are grateful to Jan Luts, an associate editor and two referees for their comments on this research.

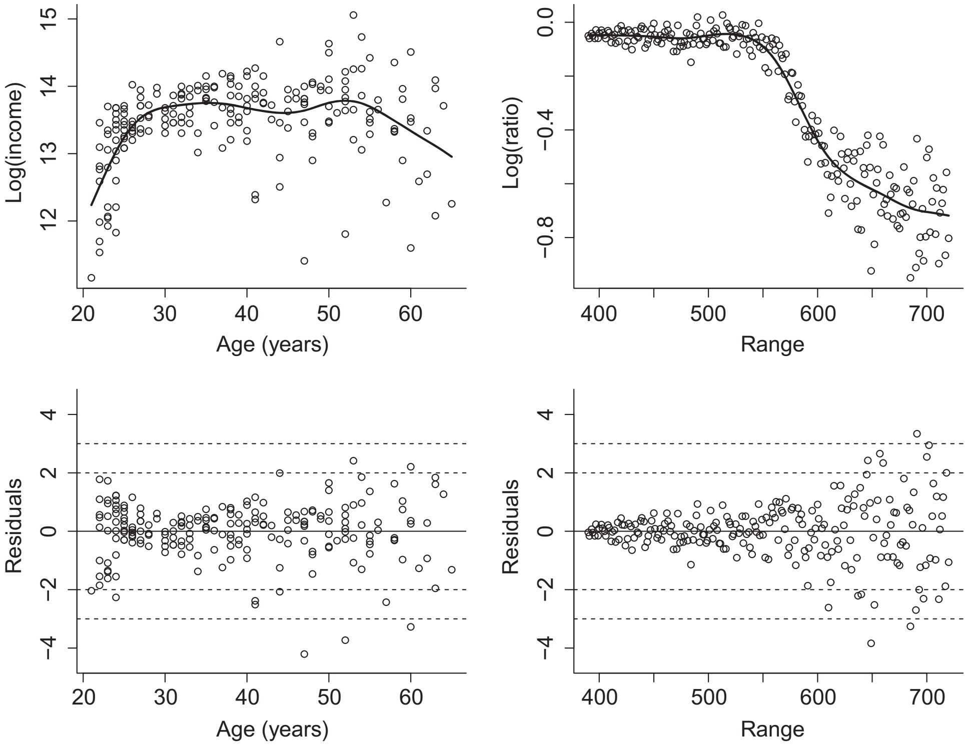

Figure 1. Two example nonparametric regression fits and corresponding standardized residual plots. Horizontal dashed lines at ±2 and ±3 aid assessment of standard normality of the standardized residuals.

Both data sets appear in Ruppert *et al.* (2003) and are first described in its Sections 5.3 and 2.7 respectively. The scatterplots and the standardized residual plots show that there is significant heteroscedasticity, which if ignored will lead to erroneous inference, for example incorrect prediction intervals.

Heteroscedastic nonparametric regression overcomes the problems apparent in Figure 1. For a set of regression data, (x_i, y_i), $1 \leqslant i \leqslant n$, it involves replacement of the homoscedastic nonparametric model

$$\mathrm{E}(y_i) = f(x_i), \quad \mathrm{Var}(y_i) = \sigma^2$$

by

$$\mathrm{E}(y_i) = f(x_i), \quad \mathrm{Var}(y_i) = g(x_i), \tag{1}$$

with the variance function g to be estimated simultaneously with the mean function f. Several approaches to fitting (1) now exist (e.g. Ruppert *et al.* 2003; Rigby & Stasinopoulos 2005; Crainiceanu *et al.* 2007). A particularly attractive approach, from an extendability standpoint, involves graphical model representations of mixed model-based penalized splines (Wand, 2009). This allows one to take advantage of the growing body of methodology and software for approximate inference in general graphical models. Graphical model-based Bayesian inference engines such as BUGS (Spiegelhalter *et al.*, 2003), Infer.NET (Minka *et al.* 2014) and Stan (Stan Development Team 2013) now accommodate a wide range of nonparametric and semiparametric regression models (e.g. Marley & Wand 2010; Luts *et al.* 2015). However, the form of the approximate inference differs markedly among the various inference engines. The engines BUGS and Stan each use relatively slow but highly accurate Markov Chain Monte Carlo (MCMC) methodology whereas Infer.NET uses faster, but less accurate, deterministic approximations such as mean field variational Bayes (MFVB). The latter type of methodology is our focus here. A relatively new modification of MFVB, non-conjugate variational message passing, is shown to be particularly useful for handling heteroscedasticity since it results in algorithms that involve only closed form algebraic expressions.

The modularity of the graphical model and MFVB approaches implies that methodology for handling heteroscedasticity in semiparametric regression applies to arbitrarily large models. We call this the *locality* property of MFVB and it is described in section 3 of Wand *et al.* (2011). If a very large graphical model includes a component where one variable is modelled as a heteroscedastic nonparametric regression function of another variable then variational inference for the parameters in that part of the graph can be achieved using the methodology developed here.

Variational methodology for heteroscedastic regression models has also been developed by Lázaro-Gredilla & Titsias (2011) and Nott, Tran & Leng (2012), although the latter article was confined to linear mean and log-variance functions. Lazaro-Gredilla & Titsias (2011) achieved simultaneous nonparametric mean and variance function estimation via an elegant Gaussian process approach. The ensuing nonparametric function estimators are full-rank and their variance function estimation strategy relies on Gauss-Hermite quadrature. Our approach differs by using low-rank penalized splines and a non-conjugate MFVB strategy that provides closed form updates, making it more amenable to high volume/velocity data.

Section 2 gives a description of the Bayesian penalized spline model for simultaneous mean and variance function estimation based on univariate data. The variational inference methodology is provided in Section 3 and our main algorithm is presented there. Section 4 provides numerical illustrations which give evidence of the speed achieved by non-conjugate MFVB. In Sections 5–7 we describe extensions to bivariate nonparametric regression, additive models and real-time semiparametric regression. Some concluding remarks are made in Section 8. The appendix contains some algebraic details used in the derivation of Algorithm 1.

2. Model description

The Gaussian heteroscedastic nonparametric regression model has generic form:

$$y_i \overset{\text{ind.}}{\sim} \mathrm{N}(f(x_i), g(x_i)), \quad 1 \leqslant i \leqslant n, \tag{2}$$

where (x_i, y_i) is the ith predictor/response pair of a regression data-set and $\overset{\text{ind.}}{\sim}$ means 'distributed independently as'. The functions f and g are smooth, but otherwise arbitrary, functions. We refer to f as the *mean function* and g as the *variance function.* We also use the term *standard deviation function* for $\sqrt{g}$.

Mixed model-based penalized spline fitting of (2) (e.g. Ruppert *et al.* 2003) involves modelling f and g according to:

$$
\begin{aligned}
&f(x)=\beta_0+\beta_1x+\sum_{k=1}^{K_u}u_kz_k^u(x), \qquad u_k\overset{\text{ind.}}{\sim}\text{N}(0,\sigma_u^2)\\
\text{and}\quad &g(x)=\exp\left(\gamma_0+\gamma_1x+\sum_{k=1}^{K_v}v_kz_k^v(x)\right), \quad v_k\overset{\text{ind.}}{\sim}\text{N}(0,\sigma_v^2).
\end{aligned} \tag{3}
$$

The $\{z_k^u : 1\leqslant k\leqslant K_u\}$ and $\{z_k^v : 1\leqslant k\leqslant K_v\}$ are spline bases of sizes K_u and K_v, respectively. Our default for the z_k^u and z_k^v are suitably transformed cubic O'Sullivan splines, as described in section 4 of Wand & Ormerod (2008). If $K_u = K_v$ then the two bases are identical, but we leave open the possibility for different basis sizes for the mean and variance functions.

In this article, we take a Bayesian approach to fitting (2) and (3), and impose the following priors on the model parameters:

$$
\beta_0,\beta_1\overset{\text{ind.}}{\sim}\text{N}(0,\sigma_\beta^2),\quad \gamma_0,\gamma_1\overset{\text{ind.}}{\sim}\text{N}(0,\sigma_\gamma^2),\quad \sigma_u\sim\text{Half-Cauchy }(A_u)\text{ and }\sigma_v\sim\text{Half-Cauchy }(A_v)
$$

with hyperparameters σ_β, σ_γ, A_u, $A_v > 0$ to be specified by the user. The Half-Cauchy (A) density function is given by $p(x) = \{2/(\pi A)\}/\{1 + (x/A)^2\}$, $x > 0$. Assuming that the data have been pre-transformed to have zero mean and unit variance, our default setting of the hyperparameters throughout this article are:

$$
\sigma_\beta=\sigma_\gamma=A_u=A_v=10^5.
$$

Result 5 of Wand *et al.* (2011) allows us to replace $\sigma_u\sim\text{Half-Cauchy}(A_u)$ by

$$
\sigma_u^2|a_u\sim\text{Inverse-Gamma}(\tfrac{1}{2},1/a_u),\quad a_u\sim\text{Inverse-Gamma}(\tfrac{1}{2},1/A_u^2), \tag{4}
$$

where $x\sim\text{Inverse-Gamma}(A,B)$ means that x has an Inverse Gamma distribution with shape parameter $A > 0$ and rate parameter $B > 0$. The corresponding density function is $p(x) = B^A\Gamma(A)^{-1}x^{-A-1}\exp(-B/x), x > 0$. Representation (4) is more amenable to variational approximate inference. A similar replacement is made for σ_v.

The full Bayesian hierarchical model corresponding to (2) is:

$$
\begin{aligned}
&\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{u},\boldsymbol{v}\sim\text{N}\left(\boldsymbol{X\beta}+\boldsymbol{Z}_u\boldsymbol{u},\text{diag}\{\exp(\boldsymbol{X\gamma}+\boldsymbol{Z}_v\boldsymbol{v})\}\right),\\
&\boldsymbol{u}|\sigma_u^2\sim\text{N}(\boldsymbol{0},\sigma_u^2\boldsymbol{I}),\quad \boldsymbol{v}|\sigma_v^2\sim\text{N}(\boldsymbol{0},\sigma_v^2\boldsymbol{I}),\quad \boldsymbol{\beta}\sim\text{N}(0,\sigma_\beta^2\boldsymbol{I}),\quad \boldsymbol{\gamma}\sim\text{N}(0,\sigma_\gamma^2\boldsymbol{I}),\\
&\sigma_u^2|a_u\sim\text{Inverse-Gamma}\left(\frac{1}{2},1/a_u\right),\quad a_u\sim\text{Inverse-Gamma}\left(\frac{1}{2},1/A_u^2\right),\\
&\sigma_v^2|a_v\sim\text{Inverse-Gamma}\left(\frac{1}{2},1/a_v\right),\quad a_v\sim\text{Inverse-Gamma}\left(\frac{1}{2},1/A_v^2\right).
\end{aligned} \tag{5}
$$

Here $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are 2×1 vectors of fixed effects containing (β_0,β_1) and (γ_0,γ_1) respectively, $\boldsymbol{u}$ is the $K_u\times 1$ vector whose entries are $u_1,\ldots, u_{K_u}$, $\boldsymbol{v}$ is a $K_v\times 1$ vector defined similarly, and σ_u^2 and σ_v^2 are variance components corresponding to $\boldsymbol{u}$ and $\boldsymbol{v}$ respectively. The design matrix $\boldsymbol{X}$ is the $n\times 2$ matrix consisting of a column of ones and a column containing the x_i's, $1\leqslant i\leqslant n$. Also, $\boldsymbol{Z}_u$ is the $n\times K_u$ matrix with (i,k) entry equal to $z_k^u(x_i)$ for $1\leqslant k\leqslant K_u$ and $\boldsymbol{Z}_v$ is the $n\times K_v$ matrix with (i,k) entry equal to $z_k^v(x_i)$ for $1\leqslant k\leqslant K_v$.

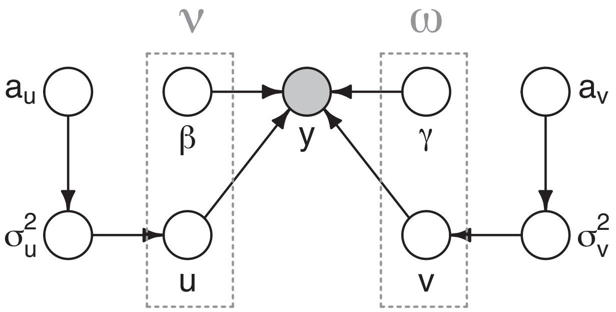

Figure 2. Directed acyclic graph for the model in (5). The shaded node corresponds to the observed data vector. Random effects and auxiliary variables are referred to as hidden nodes. The grey dashed boxes indicate that $\boldsymbol{\beta}$ and $\boldsymbol{u}$ are combined into a vector denoted by $\boldsymbol{\nu}$ and $\boldsymbol{\omega}$ is the concatenation of $\boldsymbol{\gamma}$ and $\boldsymbol{v}$.

It is also convenient to combine the mean function coefficients and variance function coefficients into single vectors:

$$\boldsymbol{\nu}\equiv\begin{bmatrix}\boldsymbol{\beta}\\ \boldsymbol{u}\end{bmatrix}\quad\text{and}\quad\boldsymbol{\omega}\equiv\begin{bmatrix}\boldsymbol{\gamma}\\ \boldsymbol{v}\end{bmatrix}.$$

An analogous combining of the design matrices: $\boldsymbol{C}_\nu\equiv[\boldsymbol{X}\ \boldsymbol{Z}_u]$, $\boldsymbol{C}_\omega\equiv[\boldsymbol{X}\ \boldsymbol{Z}_v]$, then allows us to write

$$\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{Z}_u\boldsymbol{u}=\boldsymbol{C}_\nu\boldsymbol{\nu}\quad\text{and}\quad\boldsymbol{X}\boldsymbol{\gamma}+\boldsymbol{Z}_v\boldsymbol{v}=\boldsymbol{C}_\omega\boldsymbol{\omega}.$$

Figure 2 shows the directed acyclic graph corresponding to the model conveyed in (5).

MCMC schemes for fitting and inference in (5) are relatively straightforward to devise and implement. The BUGS and Stan MCMC-based inference engines also support (5) and illustration of this fact is given in Section 4. However, MCMC does not scale well to high volume/velocity data and larger models with heteroscedastic nonparametric regression components.

3. Variational inference methodology

We now consider the problem of mean field-type variational inference for (5) (e.g. Wainwright & Jordan 2008; Ormerod & Wand 2010). This involves an approximation to the joint posterior density function of the form

$$p(\boldsymbol{\nu},\boldsymbol{\omega},\boldsymbol{\sigma}^2,\boldsymbol{a}|\boldsymbol{y})\approx q(\boldsymbol{\nu})\,q(\boldsymbol{\omega})\,q(\boldsymbol{\sigma}^2)\,q(\boldsymbol{a})\tag{6}$$

where $\boldsymbol{\sigma}^2=(\sigma_u^2,\sigma_v^2)$ and $\boldsymbol{a}=(a_u,a_v)$. The q-densities are chosen to minimize the Kullback–Leibler divergence between the left-hand-side and right-hand-side of (6). This is equivalent to maximizing the variational lower bound on the marginal log-likelihood $\log p(\boldsymbol{y})$:

$$\log\underline{p}(\boldsymbol{y};q)\equiv\mathrm{E}_q\left[\log\left\{p(\boldsymbol{\nu},\boldsymbol{\omega},\boldsymbol{\sigma}^2,\boldsymbol{a},\boldsymbol{y})\right\}-\log\{q(\boldsymbol{\nu})\,q(\boldsymbol{\omega})\,q(\boldsymbol{\sigma}^2)\,q(\boldsymbol{a})\right\}\Big].$$

The optimal q densities, denoted by q^*, can be shown to satisfy (e.g. Ormerod & Wand 2010):

$$\begin{aligned} &q^*(\boldsymbol{\nu}) \propto \exp\left\{\mathrm{E}_{q(-\boldsymbol{\nu})} \log p(\boldsymbol{\nu}|\text{rest})\right\}, \quad q^*(\boldsymbol{\omega}) \propto \exp\left\{\mathrm{E}_{q(-\boldsymbol{\omega})} \log p(\boldsymbol{\omega}|\text{rest})\right\}, \\ &q^*(\boldsymbol{\sigma}^2) \propto \exp\left\{\mathrm{E}_{q(-\boldsymbol{\sigma}^2)} \log p(\boldsymbol{\sigma}^2|\text{rest})\right\} \quad \text{and} \quad q^*(\boldsymbol{a}) \propto \exp\left\{\mathrm{E}_{q(-\boldsymbol{a})} \log p(\boldsymbol{a}|\text{rest})\right\}, \end{aligned} \tag{7}$$

where, for example $\mathrm{E}_{q(-\boldsymbol{\nu})}$ denotes expectation with respect to the q-densities of all parameters except $\boldsymbol{\nu}$. Also 'rest' denotes all of the random variables in the model other than those in $\boldsymbol{\nu}$, including $\boldsymbol{y}$.

The full conditionals $p(\boldsymbol{\nu}|\text{rest})$, $p(\boldsymbol{\sigma}^2|\text{rest})$ and $p(\boldsymbol{a}|\text{rest})$ each have standard forms which result in closed form q-density expressions. However, $p(\boldsymbol{\omega}|rest)$ is a non-standard form and challenging integrals arise in the determination of $q^*(\boldsymbol{\omega})$. We achieve a tractable solution by imposing the additional restriction that

$$q(\boldsymbol{\omega}) = q(\boldsymbol{\omega}; \boldsymbol{\mu}_{q(\boldsymbol{\omega})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}) \text{ is a N}\left(\boldsymbol{\mu}_{q(\boldsymbol{\omega})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}\right) \text{ density function} \tag{8}$$

for some mean vector $\boldsymbol{\mu}_{q(\boldsymbol{\omega})}$ and covariance matrix $\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}$. The corresponding marginal log-likelihood is

$$\begin{aligned} \log \underline{p}(\boldsymbol{y}; q, \boldsymbol{\mu}_{q(\boldsymbol{\omega})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}) \equiv \mathrm{E}_q\Big[&\log p(\boldsymbol{\nu}, \boldsymbol{\omega}, \boldsymbol{\sigma}^2, \boldsymbol{a}, \boldsymbol{y}) \\ &- \log\left\{q(\boldsymbol{\nu}) q(\boldsymbol{\omega}; \boldsymbol{\mu}_{q(\boldsymbol{\omega})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}) q(\boldsymbol{\sigma}^2) q(\boldsymbol{a})\right\}\Big] \end{aligned} \tag{9}$$

and minimization of the Kullback–Leibler divergence corresponds to $\boldsymbol{\mu}_{q(\boldsymbol{\omega})}$ and $\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}$ being chosen to maximise (9). This approach, due to Knowles & Minka (2011), has been labelled *non-conjugate variational message passing* since it provides a way of circumventing non-conjugacies in MFVB. (Note that *variational message passing* is an alternative formulation of MFVB, see e.g. Minka & Winn 2008.) Knowles & Minka (2011) propose a fixed point iteration scheme as a means of maximizing (9). Wand (2014) provides the algebraic details and simplification of fixed point updates for the Multivariate Normal q-density parameters, such as (8).

For fixed $(\boldsymbol{\mu}_{q(\boldsymbol{\omega})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})})$, application of (7), with $q^*(\boldsymbol{\omega})$ omitted, leads to:

$$\begin{aligned} &q^*(\boldsymbol{\nu}) \text{ is the N } \left(\boldsymbol{\mu}_{q(\boldsymbol{\nu})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\nu})}\right) \text{density function,} \quad q^*(\boldsymbol{\sigma}^2) = q^*(\sigma_u^2, \sigma_v^2) \text{ is a product} \\ &\text{of Inverse-Gamma} \left(\frac{1}{2}(K_u+1), B_{q(\sigma_u^2)}\right) \text{ and Inverse-Gamma} \left(\frac{1}{2}(K_v+1), B_{q(\sigma_v^2)}\right) \\ &\text{density functions, and } q^*(\boldsymbol{a}) = q^*(a_u, a_v) \text{ is a product of} \\ &\text{Inverse-Gamma}(1, B_{q(a_u)}) \text{ and Inverse-Gamma}(1, B_{q(a_v)}) \text{ density functions} \end{aligned} \tag{10}$$

for parameters $\boldsymbol{\mu}_{q(\boldsymbol{\nu})}$ and $\boldsymbol{\Sigma}_{q(\boldsymbol{\nu})}$, the mean and covariance matrix of $q^*(\boldsymbol{\nu})$, $B_{q(\sigma_u^2)}$, the rate parameter of $q^*(\sigma_u^2)$, $B_{q(\sigma_v^2)}$, the rate parameter of $q^*(\sigma_v^2)$, $B_{q(a_u)}$, the rate parameter of $q^*(a_u)$ and $B_{q(a_v)}$, the rate parameter of $q^*(a_v)$. These optimal parameters are all inter-related and obtained through an iterative algorithm, listed below as Algorithm 1. The algorithm also includes fixed point iterative updates for $\boldsymbol{\mu}_{q(\boldsymbol{\omega})}$ and $\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}$. Details on the derivation of (10), as well as the fixed point updates, are given in an Appendix.

Before presenting Algorithm 1 some additional notation is required. We use ← to denote assignment, as used in the R programming environment. For two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of equal size, $\boldsymbol{A} \odot \boldsymbol{B}$ denotes their element-wise product. If $\boldsymbol{a}$ is a $d \times 1$ vector then $\mathrm{diag}(\boldsymbol{a})$

is the $d \times d$ diagonal matrix with the entries of $\boldsymbol{a}$ along the diagonal. For a $d \times d$ matrix $\boldsymbol{A}$, diagonal($\boldsymbol{A}$) is the $d \times 1$ vector comprising the diagonal entries of $\boldsymbol{A}$. Also, $\boldsymbol{\mu}_{q(\boldsymbol{u})}$ is defined to be the sub-vector of $\boldsymbol{\mu}_{q(\boldsymbol{v})}$ corresponding to $\boldsymbol{u}$, $\boldsymbol{\Sigma}_{q(\boldsymbol{u})}$ is the sub-matrix of $\boldsymbol{\Sigma}_{q(\boldsymbol{v})}$ corresponding to $\boldsymbol{u}$. The symbols $\boldsymbol{\mu}_{q(\boldsymbol{v})}$ and $\boldsymbol{\Sigma}_{q(\boldsymbol{v})}$ are defined similarly.

Algorithm 1. MFVB algorithm for the determination of the optimal parameters in $q^*(\boldsymbol{\omega})$, $q^*(\boldsymbol{v})$, $q^*(\sigma_u^2)$, $q^*(\sigma_v^2)$, $q^*(a_u)$ and $q^*(a_v)$.

Initialize: $\boldsymbol{\mu}_{q(\boldsymbol{\omega})}$ a $(K_v+2) \times 1$ vector, $\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}$ a $(K_v+2) \times (K_v+2)$ positive definite matrix, $\boldsymbol{\mu}_{q(1/\sigma_u^2)}, \boldsymbol{\mu}_{q(1/\sigma_v^2)} > 0$, and $\boldsymbol{\mu}_{q(r_{\boldsymbol{v}}^2)}$ an $n \times 1$ vector.
Cycle:

$$\boldsymbol{\psi}_{q(\boldsymbol{\omega})} \leftarrow \exp\left\{-\boldsymbol{C}_\omega \boldsymbol{\mu}_{q(\boldsymbol{\omega})} + \frac{1}{2}\text{diagonal}\left(\boldsymbol{C}_\omega \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})} \boldsymbol{C}_\omega^\top\right)\right\}$$

$$\boldsymbol{\Sigma}_{q(\boldsymbol{v})} \leftarrow \left(\boldsymbol{C}_v^\top \text{diag}(\boldsymbol{\psi}_{q(\boldsymbol{\omega})})\boldsymbol{C}_v + \begin{bmatrix} \sigma_\beta^{-2}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mu}_{q(1/\sigma_u^2)}\boldsymbol{I} \end{bmatrix}\right)^{-1}$$

$$\boldsymbol{\mu}_{q(\boldsymbol{v})} \leftarrow \boldsymbol{\Sigma}_{q(\boldsymbol{v})} \boldsymbol{C}_v^\top \text{diag}(\boldsymbol{\psi}_{q(\boldsymbol{\omega})})\boldsymbol{y}$$

$$\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})} \leftarrow \left\{\boldsymbol{C}_\omega^\top \text{diag}\left(\boldsymbol{\mu}_{q(r_{\boldsymbol{v}}^2)} \odot \boldsymbol{\psi}_{q(\boldsymbol{\omega})}\right) \boldsymbol{C}_\omega + \begin{bmatrix} \sigma_\gamma^{-2}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mu}_{q(1/\sigma_v^2)}\boldsymbol{I} \end{bmatrix}\right\}^{-1}$$

$$\boldsymbol{\mu}_{q(\boldsymbol{\omega})} \leftarrow \boldsymbol{\mu}_{q(\boldsymbol{\omega})} + \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}\left\{\boldsymbol{C}_\omega^\top\left(\boldsymbol{\mu}_{q(r_{\boldsymbol{v}}^2)} \odot \boldsymbol{\psi}_{q(\boldsymbol{\omega})} - \boldsymbol{1}\right) - \begin{bmatrix} \sigma_\gamma^{2}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mu}_{q(1/\sigma_{\boldsymbol{v}}^2)}\boldsymbol{I} \end{bmatrix} \boldsymbol{\mu}_{q(\boldsymbol{\omega})}\right\}$$

$$\mu_{q(1/a_u)} \leftarrow 1/(\mu_{q(1/\sigma_{\boldsymbol{u}}^2)} + A_u^2) \quad ; \quad \mu_{q(1/a_{\boldsymbol{v}})} \leftarrow 1/(\mu_{q(1/\sigma_{\boldsymbol{v}}^2)} + A_v^2)$$

$$\mu_{q(1/\sigma_{\boldsymbol{u}}^2)} \leftarrow \frac{K_u + 1}{2\mu_{q(1/a_{\boldsymbol{u}})} + \|\boldsymbol{\mu}_{q(\boldsymbol{u})}\|^2 + \mathbf{tr}\left(\boldsymbol{\Sigma}_{q(\boldsymbol{u})}\right)}$$

$$\mu_{q(1/\sigma_{\boldsymbol{v}}^2)} \leftarrow \frac{K_v + 1}{2\mu_{q(1/a_{\boldsymbol{v}})} + \|\boldsymbol{\mu}_{q(\boldsymbol{v})}\|^2 + \text{tr}\left(\boldsymbol{\Sigma}_{q(\boldsymbol{v})}\right)}$$

$$\boldsymbol{\mu}_{q(r_{\boldsymbol{v}}^2)} \leftarrow \text{diagonal}\left\{\left(\boldsymbol{y} - \boldsymbol{C}_v\boldsymbol{\mu}_{q(\boldsymbol{v})}\right)\left(\boldsymbol{y} - \boldsymbol{C}_v\boldsymbol{\mu}_{q(\boldsymbol{v})}\right)^\top + \boldsymbol{C}_v \boldsymbol{\Sigma}_{q(\boldsymbol{v})} \boldsymbol{C}_v^\top\right\}$$

until the absolute relative change in $\log \underline{p}(\boldsymbol{y}; q, \boldsymbol{\mu}_{q(\boldsymbol{\omega})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})})$ is negligible.
The symbol ← indicates 'is assigned the value' or 'is replaced by'.

Convergence of Algorithm 1 can be monitored using the following explicit expression for the marginal log-likelihood lower bound:

$$\begin{aligned}\log \underline{p}(\boldsymbol{y}; q, \boldsymbol{\mu}_{q(\boldsymbol{\omega})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}) = &\frac{1}{2}\left(K_u + K_v + 4\right) - \frac{n}{2}\log(2\pi) + \log\Gamma\left(\frac{1}{2}(K_u+1)\right) \\ &+ \log\Gamma\left(\frac{1}{2}(K_v+1)\right) - 2\log(\pi) - \log(A_u) - \log(A_v) \\ &- \frac{n}{2}\boldsymbol{1}^\top(\boldsymbol{C}_\omega\boldsymbol{\mu}_{q(\boldsymbol{\omega})}) - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{C}_v\boldsymbol{\mu}_{q(\boldsymbol{v})})^\top \text{diag}(\boldsymbol{\psi}_{q(\boldsymbol{\omega})})(\boldsymbol{y} - \boldsymbol{C}_v\boldsymbol{\mu}_{q(\boldsymbol{v})}) \\ &- \frac{1}{2}\text{tr}\left(\boldsymbol{C}_v^\top\boldsymbol{C}_v\boldsymbol{\Sigma}_{q(\boldsymbol{v})}\right) - \log(\sigma_\beta^2) - \log(\sigma_\gamma^2) + \frac{1}{2}\log|\Sigma_{q(\boldsymbol{v})}| \\ &+ \frac{1}{2}\log|\Sigma_{q(\boldsymbol{\omega})}| - \frac{1}{2\sigma_\beta^2}\left(\|\boldsymbol{\mu}_{q(\boldsymbol{\beta})}\|^2 + \text{tr}(\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})})\right)\end{aligned}$$

$$
\begin{aligned}
&-\frac{1}{2\sigma_\gamma^2}\left\{\|\boldsymbol{\mu}_{q(\boldsymbol{\gamma})}\|^2+\mathrm{tr}(\boldsymbol{\Sigma}_{q(\boldsymbol{\gamma})})\right\}-\frac{1}{2}(K_u+1)\log(B_{q(\sigma_u^2)})\\
&-\frac{1}{2}(K_v+1)\log(B_{q(\sigma_v^2)})-\log\left(\mu_{q(1/\sigma_u^2)}+A_u^{-2}\right)\\
&-\log\left(\mu_{q(1/\sigma_v^2)}+A_v^{-2}\right)+\mu_{q(1/\sigma_u^2)}\mu_{q(1/a_u)}+\mu_{q(1/\sigma_v^2)}\,\mu_{q(1/a_v)}.
\end{aligned}
$$

Unlike ordinary MFVB, there is no guarantee that each iteration will lead to an increase in $\underline{p}(\boldsymbol{y};\, q, \boldsymbol{\mu}_{q(\boldsymbol{\omega})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})})$. Thus the iterations should be stopped when the absolute relative change in its logarithm falls below a negligible amount.

In Figure 3 we return to the example of Figure 1, armed with our new MFVB methodology. The top panels show the fitted mean functions, according to (5), with pointwise 95% credible sets. The model was fitted using MFVB corresponding to Algorithm 1 and MCMC based on BUGS (Spiegelhalter *et al.* 2003) with a burn-in of 5000, kept sample of 5000 and thinning factor of 5. The much faster MFVB fit is seen to be in excellent agreement with its more computationally costly benchmark. The middle panels show the fitted standard deviation functions, corresponding to $\sqrt{g}$ in the notation of (2), and corresponding 95% credible sets. The agreement between MFVB and MCMC is good, rather than excellent, for $\sqrt{g}$. The $\sqrt{g}$-standardized residual plots at the bottom of Figure 3 indicate proper accounting for the heteroscedasticity and agreement with the normality.

We also conducted a comprehensive check in order to confirm that the relative change in the approximate marginal log-likelihood does not lead to early stopping. Over several hundred runs, with data generated from different scenarios, we recorded Bayes estimates of f and g at the stopping point and again with iterations continuing 25% beyond the stopping point. The differences in the estimates were negligible.

4. Performance assessment

We conducted a comprehensive simulation study to assess the performance of Algorithm 1 in terms of inferential accuracy and computing time. Data were simulated according to model (2) with $n = 500$ and the x_i's uniform on $(0, 1)$. The four mean and variance function pairs are listed in Table 1, where $\phi(\cdot;\mu,\sigma)$ and $\Phi(\cdot;\mu,\sigma)$ respectively denote the density and distribution functions of the Normal distribution with mean μ and standard deviation σ. We generated 100 data-sets for each of the functions shown in Table 1.

For each model corresponding to a new replication, inference was achieved using MFVB via Algorithm 1 and MCMC for comparison. The R programming environment was used for implementation of Algorithm 1. The MFVB iterations were terminated when the relative change in $\log\underline{p}(\boldsymbol{y}; q)$ fell below 10^{-7}. The MCMC procedure was performed using BUGS with sample sizes the same as for the example in Section 3. The following sections provide details on the accuracy of MFVB against the MCMC benchmark.

4.1. Assessment of accuracy

Algorithm 1 yields fast approximate inference for the model parameters, however it does not guarantee that an adequate level of accuracy will be achieved. Figure 4 provides an accuracy assessment of Algorithm 1 using side-by-side boxplots of the accuracy scores for the parameters of interest.

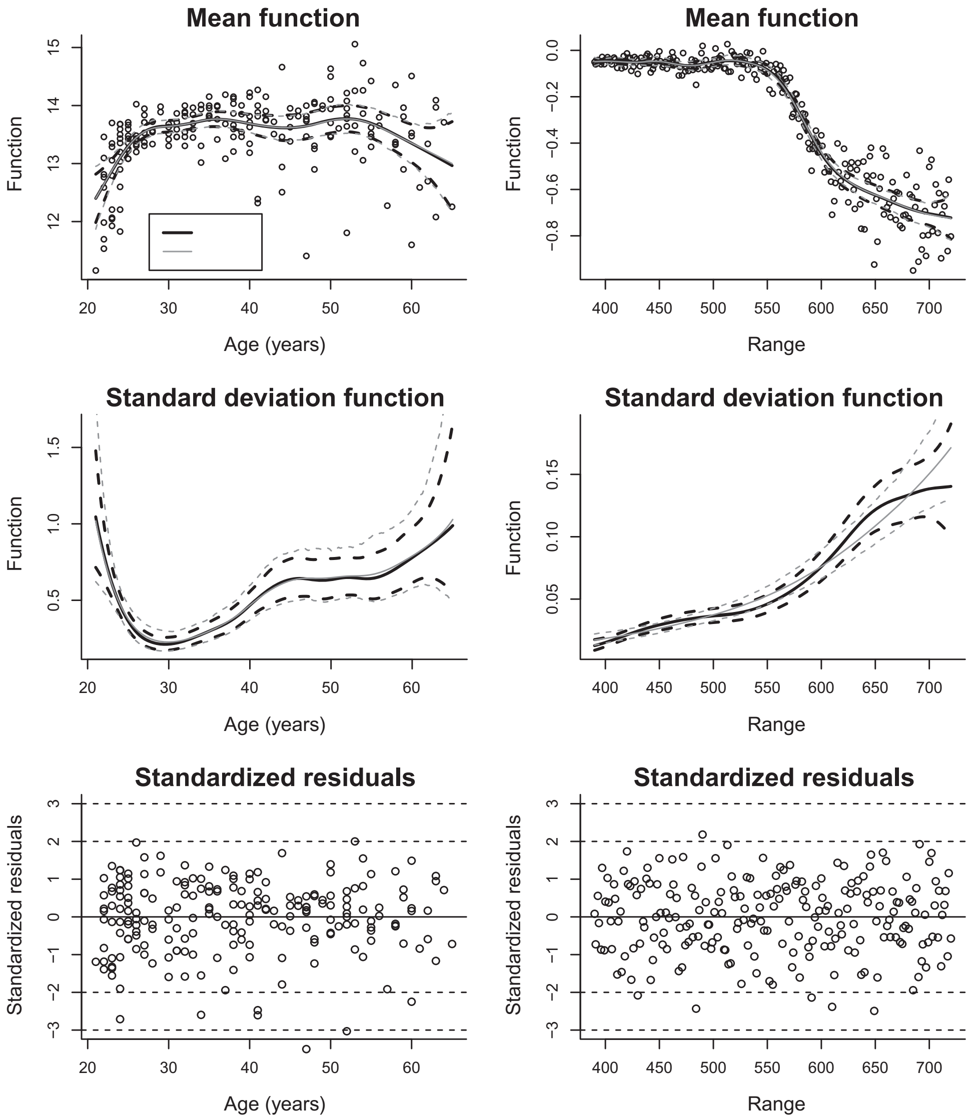

Figure 3. Top panels: fitted mean functions for two example data sets from Wand & Jones (1995) and Ruppert *et al.* (2003). The solid curves are approximate pointwise posterior means whilst the dashed curves are corresponding pointwise 95% credible sets. The approximate fits are based on MFVB via Algorithm 1 and MCMC via BUGS. Middle panels: similar to top panels but for the standard deviation function. Bottom panels: standardized residual plots based on $\{y - \widehat{f}(x_i)\}/\sqrt{\{\widehat{g}(x_i)\}}$ where $\widehat{f}$ and $\widehat{g}$ are the MFVB-approximate Bayes estimates of f and g.

For a generic parameter θ, the accuracy of $q^*(\theta)$ is defined to be

$$\text{accuracy}(q^*) = 100\left(1 - \frac{1}{2}\int_{-\infty}^{\infty} \left|q^*(\theta) - p(\theta|\mathbf{y})\right| d\theta\right)\%.$$

TABLE 1

Details of simulation study settings.

Setting	$f(x)$	*log* $g(x)$
A	$\sin(3\pi x^2)$	$0.1 + \cos(4\pi x)$
B	$-1.02x + 0.018x^2 + 0.4\phi(x; 0.38, 0.08)$ $+ 0.08\phi(x; 0.75, 0.03)$	$-0.5 - \Phi(x; 0.2, 0.1) + 0.3x^2$
C	$0.35\phi(x; 0.01, 0.08) + 1.9\phi(x; 0.45, 0.23)$ $+ 1.8\{1 - \phi(x; 0.7, 0.14)\}$	$0.3\phi(x; 0, 0.2) + 0.4\phi(x; 1, 0.1)$
D	$\sin(3\pi x^2) - 1.02x + 0.018x^2$ $+ 0.4\phi(x; 0.38, 0.08)$	$\cos(4\pi x) - 0.4 + 0.3x^2$ $- \Phi(x; 0.2, 0.1)$

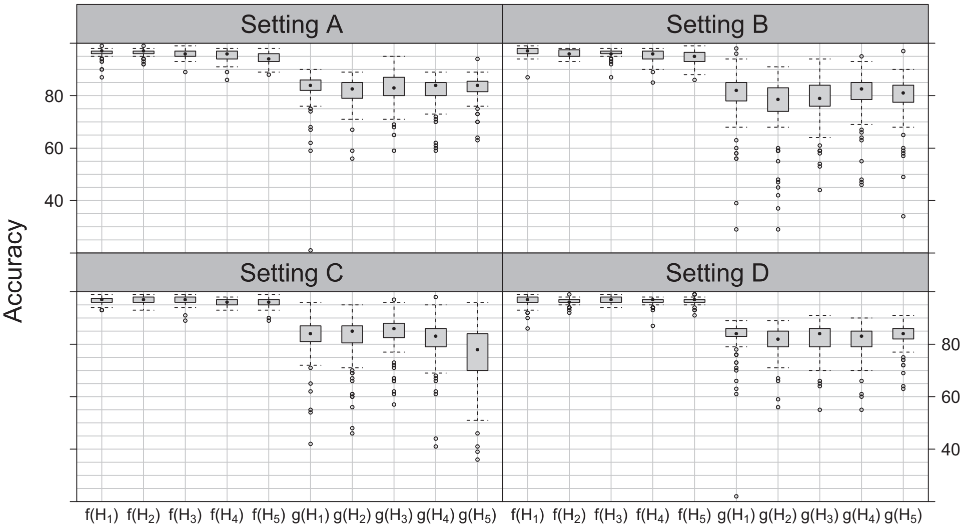

Figure 4. Summary of simulation study where the accuracy values are summarized as a boxplot.

Note that $p(\theta|\mathbf{y})$ can be approximated by using kernel density estimation applied to the MCMC sample. This accuracy score is justified in Faes *et al.* (2011). Accuracy is monitored for parameters $f(H_k)$ and $g(H_k)$, $1 \leqslant k \leqslant 5$, where the H_k are the sample hexiles of the x_i's. The boxplots illustrate that most of the accuracies for the $f(H_k)$ lie around 90%, while accuracies for the $g(H_k)$ lie around 80%. These very good accuracy results are in keeping with the heuristics given in section 3.1 of Menictas & Wand (2013).

Figure 5 shows a comparison of MCMC and the MFVB fitted mean and standard deviation functions for the first replication in each of the four simulation settings. The MCMC and MFVB fits show excellent agreement for the mean functions and relatively good agreement for the standard deviation functions.

4.2. Assessment of coverage

We are also interested in the comparison between the coverage gained by the MFVB approximate credible intervals and the true coverage. Table 2 gives the percentages of the true parameter coverage based on the approximate 95% credible intervals attained from the

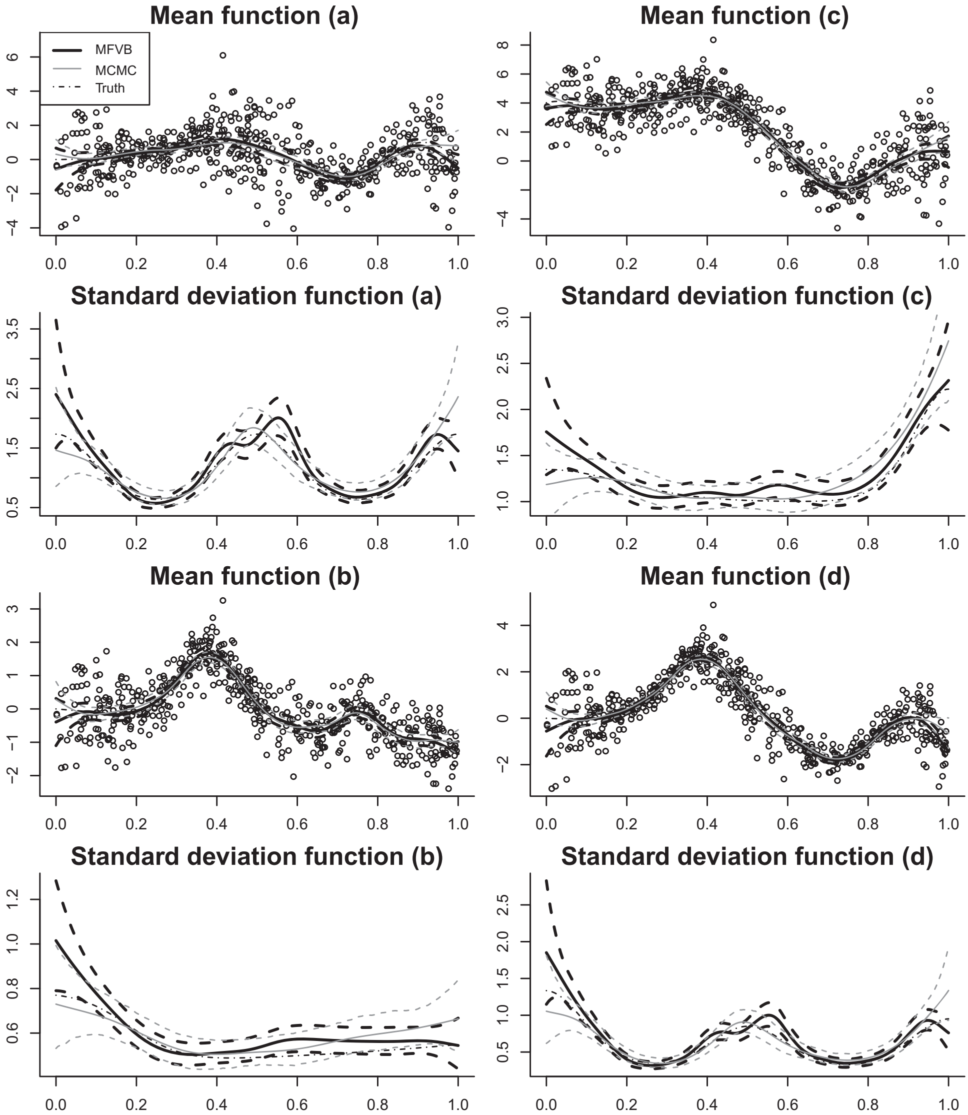

Figure 5. Comparison of MCMC and MFVB fitted functions for the first data-set in each of the four simulation settings.

MFVB posterior densities. The coverage overall is good and does not fall below 86%. As we have already seen in the previous section the performance of MFVB is excellent for the mean function and very good for the variance function.

4.3. Assessment of speed

During the running of the simulation we monitored the time taken per model to be fitted via MCMC and MFVB. The results are summarized in Table 3. The simulation was run on the first author's desktop computer (Intel Core i5-2400 3.10 GHz processor, 8 GBytes of random access memory).

TABLE 2

Percentage coverage of the true parameter values by approximate 95% credible intervals based on variational Bayes approximate posterior density functions. The percentages are based on 100 replications.

	$f(H_1)$	$f(H_2)$	$f(H_3)$	$f(H_4)$	$f(H_5)$	$g(H_1)$	$g(H_2)$	$g(H_3)$	$g(H_4)$	$g(H_5)$
sett. A	98	98	94	98	97	89	87	83	87	82
sett. B	98	98	95	96	95	83	83	90	89	83
sett. C	99	98	96	99	99	90	91	92	90	76
sett. D	98	98	95	98	98	89	89	83	86	82

TABLE 3

99% Wilcoxon confidence intervals based on run times in seconds for MCMC and MFVB fitting.

	MCMC	MFVB
setting A	(1225.49, 1228.45)	(1.69, 1.87)
setting B	(1232.96, 1238.53)	(4.95, 5.89)
setting C	(1185.45, 1188.01)	(3.29, 4.67)
setting D	(1217.77, 1364.56)	(1.26, 1.38)

As mentioned previously, convergence was assessed differently for the two approaches. Also, the speed gains of MFVB are traded off against accuracy losses which are invoked by the product restriction given in (6). However, despite this concern, the results show that MFVB is approximately 200 times faster than MCMC when comparing all models. Thus we can assert that a model that takes minutes to run using MCMC, will take only seconds to run using our MFVB algorithm.

5. Extension to bivariate predictors

Algorithm 1 is relatively easy to extend to bivariate predictor nonparametric regression, which is closely related to *geostatistics* (e.g. Cressie 1993), which we briefly describe here. In this case, the data are of the form

$$(\boldsymbol{x}_i, y_i), \quad \boldsymbol{x}_i \in \mathbb{R}^2,\, y_i \in \mathbb{R}. \tag{11}$$

In the classical geostatistics scenario, the $\boldsymbol{x}_i$'s specify geographical locations. However, in (11) the $\boldsymbol{x}_i$'s could also represent pairs of non-geographical measurements.

The bivariate analogue of (2) is

$$y_i \overset{\text{ind.}}{\sim} \mathrm{N}(f(\boldsymbol{x}_i), g(\boldsymbol{x}_i)), \quad 1 \leqslant i \leqslant n, \tag{12}$$

where f and g are real-valued functions on $\mathbb{R}^2$. The extension of (3) is

$$\begin{aligned} &f(\boldsymbol{x}) = \beta_0 + \boldsymbol{\beta}_1^\top \boldsymbol{x} + \sum_{k=1}^{K_u} u_k\, z_k^u(\boldsymbol{x}), \quad u_k \overset{\text{ind.}}{\sim} \mathrm{N}(0, \sigma_u^2) \\ \text{and} \quad &g(\boldsymbol{x}) = \exp\left(\gamma_0 + \boldsymbol{\gamma}_1^\top \boldsymbol{x} + \sum_{k=1}^{K_v} v_k\, z_k^v(\boldsymbol{x})\right), \quad v_k \overset{\text{ind.}}{\sim} \mathrm{N}(0, \sigma_v^2), \end{aligned} \tag{13}$$

where each of $\boldsymbol{\beta}_1$ and $\boldsymbol{\gamma}_1$ are 2×1 vectors. The functions $\{z_k^u : 1 \leqslant k \leqslant K_u\}$ and $\{z_k^v : 1 \leqslant k \leqslant K_v\}$ are now bivariate spline basis functions. A reasonable default for the z_k^u (Ruppert *et al.* 2003) is the low-rank thin plate spline basis with kth element:

$$z_k^u(\boldsymbol{x}) = r(\|\boldsymbol{x} - \boldsymbol{\kappa}_k^u\|)[r(\|\boldsymbol{\kappa}_k^u - \boldsymbol{\kappa}_{k'}^u\|)]^{-1/2}_{1 \leqslant k,k' \leqslant K_u}, \tag{14}$$

where $\boldsymbol{\kappa}_1^u, \ldots, \boldsymbol{\kappa}_{K_u}^u$ is a set of bivariate knot locations that efficiently cover the space of the $\boldsymbol{x}_i$'s and $r(x) \equiv x^2 \log(x)$. The default z_k^vs have an analogous definition.

According to this set-up, the only difference between the univariate nonparametric heteroscedastic regression model, treated in Section 3, and its bivariate counterpart is the basis functions and their coefficients. Hence, Algorithm 1 can be used to fit the bivariate nonparametric heteroscedastic regression model by replacing $\boldsymbol{\nu}$, $\boldsymbol{\omega}$, $\boldsymbol{C}_\nu$ and $\boldsymbol{C}_\omega$ from Section 3 with

$$\boldsymbol{\nu} \equiv \begin{bmatrix} \beta_0 \\ \boldsymbol{\beta}_1 \\ \boldsymbol{u} \end{bmatrix}, \ \boldsymbol{\omega} \equiv \begin{bmatrix} \gamma_0 \\ \boldsymbol{\gamma}_1 \\ \boldsymbol{v} \end{bmatrix}, \ \boldsymbol{C}_\nu \equiv \begin{bmatrix} 1 \ \boldsymbol{x}_i^\top \ \underset{1 \leqslant k \leqslant K^u}{z_k^u(\boldsymbol{x}_i)} \end{bmatrix}_{1 \leqslant i \leqslant n} \text{ and } \boldsymbol{C}_\omega \equiv \begin{bmatrix} 1 \ \boldsymbol{x}_i^\top \ \underset{1 \leqslant k \leqslant K^v}{z_k^v(\boldsymbol{x}_i)} \end{bmatrix}_{1 \leqslant i \leqslant n}.$$

We fitted (13) to geo-referenced data on sea-floor sediment pollution in the North Sea (source: Pebesma & Duin 2005). The data are stored in the `pcb` data-frame within the R package `gstat` (Pebesma 2004). The response variable is a measurement of polychlorinated biphenyl with Ballschmiter-Zell congener number 138 (PCB-138). The motivating study is concerned with spatial and temporal variability of PCB-138. For the purposes of illustration, we ignore the temporal aspect and focus on geographical variability in the mean and variance of the response. In the notation of model (12)–(13), the variables are $\boldsymbol{x} = (x_1, x_2)$ where

$x_1 = x$-coordinate in the Universal Mobile Telecommunications System for Zone 31,
$x_2 = y$-coordinate in the Universal Mobile Telecommunications System for Zone 31, and
$y =$ PCB-138 measured on the sediment fraction smaller than 63 parts per million, in μg/kg dry matter.

The sample size is $n = 216$ and $K_u = K_v = 50$ thin plate basis splines were used for each functional fit. The estimated mean and standard deviation functions are shown in Figure 6. Both functions are seen to exhibit pronounced spatial effects. Simple bivariate predictor models that ignore the heteroscedasticity described by the right panel of Figure 6 would lead to erroneous prediction intervals.

Higher dimensional heteroscedastic nonparametric regression can be achieved via Algorithm 1 with little notational change from the bivariate case treated here. The only required modification involves higher-dimensional thin plate spline basis functions instead of those given by (14).

6. Extension to additive models

The final semiparametric regression extension, that we discuss briefly in this section, is additive models with multiplicative variance functions. Models of this type have a small literature with Rigby & Stasinopoulos (2005) being a key reference. Their generic form is

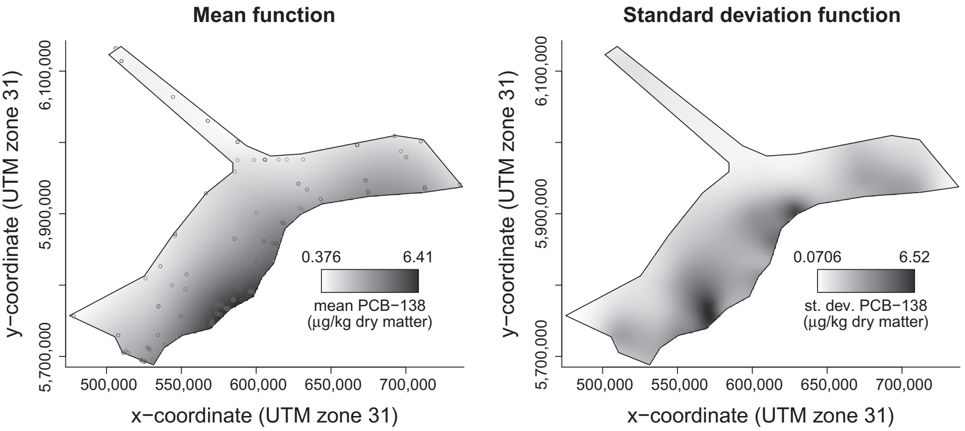

Figure 6. Left panel: Fitted mean function for the polychlorinated biphenyl data described in the text, based on MFVB via Algorithm 1. Right panel: similar to top panel but for the standard deviation function.

$$y_i \sim \text{N}\left(\beta_0 + \sum_{j=1}^{d} f_j(x_{ji}),\ \exp\left(\gamma_0 + \sum_{j=1}^{d} h_j(x_{ji})\right)\right), \quad 1 \leqslant i \leqslant n. \tag{15}$$

Here f_j and h_j, $1 \leqslant j \leqslant d$, are smooth but otherwise arbitrary functions. We use penalized spline models of the form

$$\begin{aligned} f_j(x) &= \beta_j x + \sum_{k=1}^{K_j^u} u_{jk} z_{jk}^u(x), \quad u_{jk} \overset{\text{ind.}}{\sim} \text{N}(0, \sigma_{uj}^2) \\ \text{and} \quad h_j(x) &= \gamma_j x + \sum_{k=1}^{K_j^v} v_{jk} z_{jk}^v(x), \quad v_{jk} \overset{\text{ind.}}{\sim} \text{N}(0, \sigma_{vj}^2). \end{aligned} \tag{16}$$

The $\{z_{jk}^u : 1 \leqslant k \leqslant K_j^u\}$, $1 \leqslant j \leqslant d$, are spline bases of sizes K_j^u, analogous to those presented in Section 2. The $\{z_{jk}^v : 1 \leqslant k \leqslant K_j^v\}$, $1 \leqslant j \leqslant d$, are similarly defined. The priors on the regression coefficients and standard deviation parameters are

$$\begin{aligned} &\beta_j \overset{\text{ind.}}{\sim} \text{N}(0, \sigma_{\boldsymbol{\beta}}^2), \quad \gamma_j \overset{\text{ind.}}{\sim} \text{N}(0, \sigma_{\boldsymbol{\gamma}}^2), \\ &\sigma_{uj} \overset{\text{ind.}}{\sim} \text{Half-Cauchy}(A_u), \quad \sigma_{vj} \overset{\text{ind.}}{\sim} \text{Half-Cauchy}(A_v). \end{aligned} \tag{17}$$

The Bayesian model given by (15)–(17) admits a closed form non-conjugate MFVB algorithm, with the regression coefficients for the full mean and variance functions each being Multivariate Normal. Details and illustration are given in Menictas (2015). Section 5.2 of Wand (2014) also contains an illustration for data from a Californian air pollution study.

The extension to additive models with parametric components is trivial. For example, if binary predictor data b_i, $1 \leqslant i \leqslant n$, are also available then the following extension of (15):

$$y_i \sim \mathrm{N}\left(\beta_0+\beta_b b_i+\sum_{j=1}^{d} f_j(x_{ji}),\ \exp\left(\gamma_0+\gamma_b b_i+\sum_{j=1}^{d} h_j(x_{ji})\right)\right), \quad 1\leqslant i\leqslant n$$

can be handled via simple additions to the design matrices and coefficient vectors. Similar comments apply to the addition of continous predictors that have purely linear impact on either the mean function or log-variance function.

7. Real-time heteroscedastic nonparametric regression

Almost all nonparametric regression methodology presented to date make the assumption that the data are processed in batch, that is, at the same time. However, some disadvantages of batch processing include the requirement that analysis must wait until the entire data set has been collected, and often the need to store the entire data set in memory. In the real-time case, the analysis is updated as each new data point is collected. This is beneficial, and sometimes essential, for both high volume and/or velocity data. In this section we present a variation of Algorithm 1 that allows for real-time heteroscedastic nonparametric regression.

Algorithm 2. MFVB algorithm for real-time determination of the optimal parameters in $q^*(\boldsymbol{\omega})$, $q^*(\boldsymbol{\nu})$, $q^*(\sigma_u^2)$, $q^*(\sigma_v^2)$, $q^*(a_u)$ and $q^*(a_v)$.

(i) Use Algorithm 1 to perform batch-based tuning runs, analogous to those described in Algorithm 2' of Luts *et al.* (2014), and determine a warm-up sample size n_{warm} for which convergence is validated.

(ii) Set $\boldsymbol{\mu}_{q(\boldsymbol{\nu})}$, $\boldsymbol{\Sigma}_{q(\boldsymbol{\nu})}$, $\boldsymbol{\mu}_{q(\boldsymbol{\omega})}$, $\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}$, $\mu_{q(1/\sigma_u^2)}$, and $\mu_{q(1/\sigma_v^2)}$ to their values obtained in the warm up batch-based tuning run with sample size n_{warm}. Next set $\boldsymbol{y}_{\text{warm}}$ to be the response vector on the first n_{warm} observations. Also set $\boldsymbol{C}_{\nu,warm}$ and $\boldsymbol{C}_{\omega,warm}$ to be the design matrices based on the first n_{warm} observations. Lastly assign $n \leftarrow n_{\text{warm}}$.

(iii) Cycle:

Read in $y_{\text{new}}(1\times 1), \boldsymbol{c}_{\nu,\text{new}}\ \{(2+K_u)\times 1\}$ and $\boldsymbol{c}_{\omega,\text{new}}\ \{(2+K_v)\times 1\}; n\leftarrow n+1$

$$\boldsymbol{C}_\nu \leftarrow \left[\boldsymbol{C}_\nu^\top \boldsymbol{c}_{\nu,\text{new}}\right]^\top; \boldsymbol{C}_\omega \leftarrow \left[\boldsymbol{C}_\omega^\top \boldsymbol{c}_{\omega,\text{new}}\right]^\top; \boldsymbol{y} \leftarrow \left[\boldsymbol{y}^\top y_{\text{new}}\right]^\top$$

$$\boldsymbol{\mu}_{q(r_{\boldsymbol{\nu}}^2)} \leftarrow \text{diagonal}\left\{\left(\boldsymbol{y}-\boldsymbol{C}_\nu\boldsymbol{\mu}_{q(\boldsymbol{\nu})}\right)\left(\boldsymbol{y}-\boldsymbol{C}_\nu\boldsymbol{\mu}_{q(\boldsymbol{\nu})}\right)^\top+\boldsymbol{C}_\nu\boldsymbol{\Sigma}_{q(\boldsymbol{\nu})}\boldsymbol{C}_\nu^\top\right\}$$

$$\boldsymbol{\psi}_{q(\boldsymbol{\omega})} \leftarrow \exp\left\{-\boldsymbol{C}_\omega\boldsymbol{\mu}_{q(\boldsymbol{\omega})}+\frac{1}{2}\text{diagonal}\left(\boldsymbol{C}_\omega\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}\boldsymbol{C}_\omega^\top\right)\right\}$$

$$\boldsymbol{\Sigma}_{q(\boldsymbol{\nu})} \leftarrow \left(\boldsymbol{C}_\nu^\top \text{diag}\left(\boldsymbol{\psi}_{q(\boldsymbol{\omega})}\right)\boldsymbol{C}_\nu+\begin{bmatrix}\sigma_\beta^{-2}\boldsymbol{I}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \mu_{q(1/\sigma_u^2)}\boldsymbol{I}_{K_u}\end{bmatrix}\right)^{-1}$$

$$\boldsymbol{\mu}_{q(\boldsymbol{\nu})} \leftarrow \boldsymbol{\Sigma}_{q(\boldsymbol{\nu})}\boldsymbol{C}_\nu^\top \text{diag}\left(\boldsymbol{\psi}_{q(\boldsymbol{\omega})}\right)\boldsymbol{y}$$

$$\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})} \leftarrow \left(\boldsymbol{C}_\omega^\top \text{diag}\left(\boldsymbol{\mu}_{q(\boldsymbol{r}_{\boldsymbol{\nu}}^2)}\odot\boldsymbol{\psi}_{q(\boldsymbol{\omega})}\right)\boldsymbol{C}_\omega+\begin{bmatrix}\sigma_\gamma^{-2}\boldsymbol{I}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \mu_{q(1/\sigma_v^2)}\boldsymbol{I}_{K_v}\end{bmatrix}\right)^{-1}$$

$$\boldsymbol{\mu}_{q(\boldsymbol{\omega})} \leftarrow \boldsymbol{\mu}_{q(\boldsymbol{\omega})}+\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}\left\{\boldsymbol{C}_\omega^\top\left(\boldsymbol{\mu}_{q(\boldsymbol{r}_{\boldsymbol{\nu}}^2)}\odot\boldsymbol{\psi}_{q(\boldsymbol{\omega})}-\boldsymbol{1}\right)-\begin{bmatrix}\sigma_\gamma^{-2}\boldsymbol{I}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \mu_{q(1/\sigma_v^2)}\boldsymbol{I}_{K_v}\end{bmatrix}\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\right\}$$

$$\mu_{q(1/a_{\boldsymbol{u}})} \leftarrow 1/\left(\mu_{q(1/\sigma^2_{\boldsymbol{u}})} + A_{\boldsymbol{u}}^{-2}\right); \mu_{q(1/a_{\boldsymbol{v}})} \leftarrow 1/\left(\mu_{q(1/\sigma^2_{\boldsymbol{v}})} + A_{\boldsymbol{v}}^{-2}\right)$$
$$\mu_{q(1/\sigma^2_{\boldsymbol{u}})} \leftarrow (K_u + 1)/\{2\mu_{q(1/a_{\boldsymbol{u}})} + \|\boldsymbol{\mu}_{q(\boldsymbol{u})}\|^2 + \text{tr}\left(\boldsymbol{\Sigma}_{q(\boldsymbol{u})}\right)\}$$
$$\mu_{q(1/\sigma^2_{\boldsymbol{v}})} \leftarrow (K_v + 1)/\{2\mu_{q(1/a_{\boldsymbol{v}})} + \|\boldsymbol{\mu}_{q(\boldsymbol{v})}\|^2 + \text{tr}\left(\boldsymbol{\Sigma}_{q(\boldsymbol{v})}\right)\}$$

until the analysis is complete or data no longer available.
The symbol ← indicates 'is assigned the value' or 'is replaced by'.

Algorithm 2 processes each new entry of $\boldsymbol{y}$, denoted by y_{new}, and its corresponding row of $\boldsymbol{C}_v$ and $\boldsymbol{C}_\omega$, denoted by $\boldsymbol{c}_{v,new}$ and $\boldsymbol{c}_{\omega,\text{new}}$, successively in real time. The starting values for the real-time procedure are determined by performing a sufficiently large batch fit. This is explained in more detail in section 2.1.1 of Luts *et al.* (2014).

The web-site `realtime-semiparametric-regression.net` features a movie that illustrates Algorithm 2 for data simulated according to setting *D*. The warm-up sample size is $n_{\text{warm}} = 500$. The link for the movie is titled `Heteroscedastic nonparametric regression` and portrays the effectiveness of real time processing for mean and standard deviation function fitting.

8. Concluding remarks

We have developed closed form algorithms for fast batch and real-time fitting and inference for a variety of heteroscedastic semiparametric regression models.

The methodology also applies to larger models courtesy of the locality property of mean field variational inference methods. The new methodology has been shown to perform very well on simulated and actual data.

Appendix A: Derivation of optimal *q*-density functions

A.1. Derivation of $q^*(\boldsymbol{v})$

First note that

$$\begin{aligned}\log q^*(\boldsymbol{v}) &= \text{E}_q\{\log p(\boldsymbol{v}|\text{rest})\} + \text{const}\\ &= -\frac{1}{2}\left[\boldsymbol{v}^\top\left(\boldsymbol{C}_v^\top \text{diag}\left\{\text{E}_q(e^{-\boldsymbol{C}_\omega\boldsymbol{\omega}})\right\}\boldsymbol{C}_v + \begin{bmatrix}\sigma_{\boldsymbol{\beta}}^{-2}\boldsymbol{I}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \mu_{q(1/\sigma_u^2)}\boldsymbol{I}_{K_u}\end{bmatrix}\right)\boldsymbol{v}\right.\\ &\qquad \left. -2\boldsymbol{v}^\top\boldsymbol{C}_v^\top \text{diag}\left\{\text{E}_q(e^{-\boldsymbol{C}_\omega\boldsymbol{\omega}})\right\}\boldsymbol{y}\right] + \text{const},\end{aligned}$$

where 'const' denotes terms not depending on the argument of q^*. The form of $q^*(\boldsymbol{v})$ follows from this and the fact that

$$\text{E}_q(e^{-\boldsymbol{C}_\omega\boldsymbol{\omega}}) = \exp\{-\boldsymbol{C}_\omega\boldsymbol{\mu}_{q(\boldsymbol{\omega})} + \frac{1}{2}\boldsymbol{C}_\omega\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}\boldsymbol{C}_\omega^\top\}.$$

Therefore

$$\boldsymbol{\mu}_{q(\boldsymbol{v})} = \boldsymbol{\Sigma}_{q(\boldsymbol{v})}\boldsymbol{C}_v^\top \text{diag}(\boldsymbol{\psi}_{q(\boldsymbol{\omega})})\boldsymbol{y}$$

and

$$\boldsymbol{\Sigma}_{q(\boldsymbol{v})} = \left(\boldsymbol{C}_v^\top \text{diag}\left\{\boldsymbol{\psi}_{q(\boldsymbol{\omega})}\right\}\boldsymbol{C}_v + \begin{bmatrix}\sigma_{\boldsymbol{\beta}}^{-2}\boldsymbol{I}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \mu_{q(1/\sigma_u^2/}\boldsymbol{I}_{K_u}\end{bmatrix}\right)^{-1}.$$

A.2. Derivation of $q^*(\sigma_u^2)$ and $q^*(\sigma_v^2)$

$$\begin{aligned}\log q^*(\sigma_u^2) &= \mathrm{E}_q\left\{\log p(\sigma_u^2|\text{rest})\right\} + \text{const}\\ &= \left\{-\frac{1}{2}(K_u+1)-1\right\}\log(\sigma_u^2) - \left(\frac{1}{2}\mathrm{E}_q\|\boldsymbol{u}\|^2 + \mu_{q(1/a_u)}\right)/\sigma_u^2 + \text{const.}\end{aligned}$$

The form of $q^*(\sigma_u^2)$ follows from the fact that

$$\mathrm{E}_q\|\boldsymbol{u}\|^2 = \|\mathrm{E}_q(\boldsymbol{u})\|^2 + \text{tr}\left\{Cov_q(\boldsymbol{u})\right\}.$$

The expression for $\mu_{q(1/\sigma_u^2)}$ follows from a suitable result for the Inverse-Gamma distribution. For example, if random variable v has an Inverse-Gamma distribution with density function $p(v) = B^A\Gamma(A)^{-1}v^{-A-1}\exp(-v/B)$, then $\mathrm{E}(1/v) = A/B$. Therefore

$$B_{q(\sigma_u^2)} = \frac{1}{2}\left\{\|\boldsymbol{\mu}_{q(\boldsymbol{u})}\|^2 + \text{tr}\left(\boldsymbol{\Sigma}_{q(\boldsymbol{u})}\right)\right\} + \mu_{q(1/a_u)}$$

and

$$\mu_{q(1/\sigma_u^2)} = \frac{1}{2}(K_u+1)/B_{q(\sigma_u^2)}.$$

The derivation of $B_{q(\sigma_v^2)}$ and $\mu_{q(1/\sigma_v^2)}$ is similar.

A.3. Derivation of $q^*(a_u)$ and $q^*(a_v)$

$$\begin{aligned}\log q^*(a_u) &= \mathrm{E}_q\{\log p(a_u|\text{rest})\} + \text{const}\\ &= (-1-1)\log(a_u) - \left(\mu_{q(1/\sigma_u^2)} + A_u^{-2}\right)/a_u + \text{const.}\end{aligned}$$

The expressions for $B_{q(a_u)}$ and $\mu_{q(1/a_u)}$ follow immediately. The derivation of $B_{q(a_v)}$, and $\mu_{q(1/a_v)}$ is similar to that just shown above.

A.4. Derivation of the $\boldsymbol{\mu}_{q(\boldsymbol{\omega})}$ and $\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}$ updates

The updates for $\boldsymbol{\mu}_{q(\boldsymbol{\omega})}$ and $\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}$ are based on maximisation of the current value of the marginal log-likelihood lower bound $\log\underline{p}(\boldsymbol{y}; q, \boldsymbol{\mu}_{q(\boldsymbol{\omega})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})})$ over these parameters using fixed point iteration. Wand (2014) shows that the updates reduce to

$$\begin{aligned}\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})} &\leftarrow \left\{-2\text{vec}^{-1}\left(\left(\mathrm{D}_{\text{vec}(\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})})}S\right)^{\top}\right)\right\}^{-1}\\ \boldsymbol{\mu}_{q(\boldsymbol{\omega})} &\leftarrow \boldsymbol{\mu}_{q(\boldsymbol{\omega})} + \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}\left(\mathrm{D}_{\boldsymbol{\mu}_{q(\boldsymbol{\omega})}}S\right)^{\top},\end{aligned}$$

where

$$S \equiv \mathrm{E}_q\left\{\log p(\boldsymbol{y}|\boldsymbol{v},\boldsymbol{\omega}) + \log p(\boldsymbol{\omega}|\sigma_v^2)\right\},$$

D denotes derivative vector, as defined in Magnus & Neudecker (1999), and vec and vec^{-1} are as defined in Wand (2014). However, a result in the appendix of Opper & Archambeau (2009) shows that an equivalent expression for the first of these updates is

$$\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})} \quad \leftarrow \quad \left(-\mathrm{H}_{\boldsymbol{\mu}_{q(\boldsymbol{\omega})}}S\right)^{-1}$$

where H denotes the Hessian matrix as defined in Magnus & Neudecker (1999). We work with this alternative form here. An explicit expression for S is

$$
\begin{aligned}
S \equiv & -\mathbf{1}^\top \boldsymbol{C}_\omega \boldsymbol{\mu}_{q(\boldsymbol{\omega})} - \boldsymbol{\mu}^\top_{q(r^2_{\boldsymbol{\nu}})} \exp\left\{-\boldsymbol{C}_\omega \boldsymbol{\mu}_{q(\boldsymbol{\omega})} + \frac{1}{2}\text{diagonal}\left(\boldsymbol{C}_\omega \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})} \boldsymbol{C}_\omega^\top\right)\right\} \\
& - \frac{1}{2}\text{tr}\left(\begin{bmatrix} \sigma_\gamma^{-2}\boldsymbol{I}_2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mu}_{q(1/\sigma_\nu^2)}\boldsymbol{I} \end{bmatrix} \left(\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\boldsymbol{\mu}^\top_{q(\boldsymbol{\omega})} + \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}\right)\right) \\
& - \left(n + \frac{K}{2} + 1\right)\log(2\pi) - \frac{K}{2}\log(\sigma_\nu^2) - \log(\sigma_\gamma^2).
\end{aligned}
$$

This leads to

$$
\begin{aligned}
d_{\boldsymbol{\mu}_{q(\boldsymbol{\omega})}} S = & -\mathbf{1}^\top \boldsymbol{C}_\omega d\boldsymbol{\mu}_{q(\boldsymbol{\omega})} \\
& + \boldsymbol{\mu}^\top_{q(r^2_{\boldsymbol{\nu}})}\text{diag}\left(\exp\left\{-\boldsymbol{C}_\omega \boldsymbol{\mu}_{q(\boldsymbol{\omega})} + \frac{1}{2}\text{diagonal}\left(\boldsymbol{C}_\omega \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})} \boldsymbol{C}_\omega^\top\right)\right\}\right) \boldsymbol{C}_\omega d\boldsymbol{\mu}_{q(\boldsymbol{\omega})} \\
& - \boldsymbol{\mu}^\top_{q(\boldsymbol{\omega})} \begin{bmatrix} \sigma_\gamma^{-2}\boldsymbol{I}_2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mu}_{q(1/\sigma_\nu^2)}\boldsymbol{I} \end{bmatrix} d\boldsymbol{\mu}_{q(\boldsymbol{\omega})} \\
= & \left(\left[\boldsymbol{\mu}_{q(r^2_{\boldsymbol{\nu}})} \odot \exp\left\{-\boldsymbol{C}_\omega \boldsymbol{\mu}_{q(\boldsymbol{\omega})} + \frac{1}{2}\text{diagonal}\left(\boldsymbol{C}_\omega \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})} \boldsymbol{C}_\omega^\top\right)\right\} - \mathbf{1}\right]^\top \boldsymbol{C}_\omega \right. \\
& \left. -\boldsymbol{\mu}^\top_{q(\boldsymbol{\omega})} \begin{bmatrix} \sigma_\gamma^{-2}\boldsymbol{I}_2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mu}_{q(1/\sigma_\nu^2)}\boldsymbol{I} \end{bmatrix}\right) d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}.
\end{aligned}
$$

Then, by Theorem 6, Chapter 5 of Magnus & Neudecker (1999),

$$
\begin{aligned}
\left(\mathsf{D}_{\boldsymbol{\mu}_{q(\boldsymbol{\omega})}} S\right)^\top = & \boldsymbol{C}_\omega^\top \left[\boldsymbol{\mu}_{q(r^2_{\boldsymbol{\nu}})} \odot \exp\left\{-\boldsymbol{C}_\omega \boldsymbol{\mu}_{q(\boldsymbol{\omega})} + \frac{1}{2}\text{diagonal}\left(\boldsymbol{C}_\omega \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})} \boldsymbol{C}_\omega^\top\right)\right\} - \mathbf{1}\right] \\
& - \begin{bmatrix} \sigma_\gamma^{-2}\boldsymbol{I}_2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mu}_{q(1/\sigma_\nu^2)}\boldsymbol{I} \end{bmatrix} \boldsymbol{\mu}_{q(\boldsymbol{\omega})}.
\end{aligned}
$$

Next,

$$
\begin{aligned}
d^2 \boldsymbol{\mu}_{q(\boldsymbol{\omega})} S = & \left(\boldsymbol{\mu}_{q(r^2_{\boldsymbol{\nu}})} \odot \exp\left[\left(d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\right)^\top \left\{-\boldsymbol{C}_\omega \boldsymbol{\mu}_{q(\boldsymbol{\omega})} + \frac{1}{2}\text{diagonal}\left(\boldsymbol{C}_\omega \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})} \boldsymbol{C}_\omega^\top\right)\right\}\right]^\top \boldsymbol{C}_\omega \right. \\
& \left. - \left(d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\right)^\top \left[\begin{bmatrix} \sigma_\gamma^{-2}\boldsymbol{I}_2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mu}_{q(1/\sigma_\nu^2)}\boldsymbol{I} \end{bmatrix}\right]\right) d\boldsymbol{\mu}_{q(\boldsymbol{\omega})} \\
= & \left[\left\{\boldsymbol{\mu}_{q(r^2_{\boldsymbol{\nu}})} \odot \left(\text{diag}\left[\exp\left\{-\boldsymbol{C}_\omega \boldsymbol{\mu}_{q(\boldsymbol{\omega})} + \frac{1}{2}\text{diagonal}\left(\boldsymbol{C}_\omega \boldsymbol{\Sigma}_{q(\boldsymbol{\omega})} \boldsymbol{C}_\omega^\top\right)\right\}\right]\right.\right. \\
& \left.\left. \times \left(\boldsymbol{C}_\omega d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\right)\right)\right\}^\top \boldsymbol{C}_\omega - \left(d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\right)^\top \left[\begin{bmatrix} \sigma_\gamma^{-2}\boldsymbol{I}_2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mu}_{q(1/\sigma_\nu^2)}\boldsymbol{I} \end{bmatrix}\right]\right] d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}.
\end{aligned}
$$

Then, using $\text{diag}(a)b = a \odot b$,

$$
\begin{aligned}
d^2\boldsymbol{\mu}_{q(\boldsymbol{\omega})}S = &\left\{\left(\operatorname{diag}\left(\boldsymbol{\mu}_{q(r_\nu^2)}\right)\operatorname{diag}\left[\exp\left\{-\boldsymbol{C}_\omega\boldsymbol{\mu}_{q(\boldsymbol{\omega})}+\frac{1}{2}\operatorname{diagonal}\left(\boldsymbol{C}_\omega\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}\boldsymbol{C}_\omega^\top\right)\right\}\right]\right.\right.\\
&\left.\left.\times\left(-\boldsymbol{C}_\omega d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\right)\right)^\top\boldsymbol{C}_\omega-\left(d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\right)^\top\begin{bmatrix}\sigma_\gamma^{-2}\boldsymbol{I}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \mu_{q(1/\sigma_\nu^2)}\boldsymbol{I}\end{bmatrix}\right\}d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\\
= &\left(d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\right)^\top\big(-\boldsymbol{C}_\omega^\top\operatorname{diag}\big[\boldsymbol{\mu}_{q(r_\nu^2)}\odot\exp\big\{-\boldsymbol{C}_\omega\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\\
&+\frac{1}{2}\operatorname{diagonal}\big(\boldsymbol{C}_\omega\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}\boldsymbol{C}_\omega^\top\big)\big\}\big]\big)\boldsymbol{C}_\omega d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\\
&-\left(d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}\right)^\top\begin{bmatrix}\sigma_\gamma^{-2}\boldsymbol{I}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \mu_{q(1/\sigma_\nu^2)}\boldsymbol{I}\end{bmatrix}d\boldsymbol{\mu}_{q(\boldsymbol{\omega})}.
\end{aligned}
$$

Therefore, by Theorem 6, Chapter 6 of Magnus & Neudecker (1999),

$$
\begin{aligned}
-\mathsf{H}_{\boldsymbol{\mu}_{q(\boldsymbol{\omega})}}S = &\,\boldsymbol{C}_\omega^\top\operatorname{diag}\left[\boldsymbol{\mu}_{q(r_\nu^2)}\odot\exp\left\{\boldsymbol{C}_\omega\boldsymbol{\mu}_{q(\boldsymbol{\omega})}+\frac{1}{2}\operatorname{diagonal}\left(\boldsymbol{C}_\omega\boldsymbol{\Sigma}_{q(\boldsymbol{\omega})}\boldsymbol{C}_\omega^\top\right)\right\}\right]\boldsymbol{C}_\omega\\
&+\begin{bmatrix}\sigma_\gamma^{-2}\boldsymbol{I}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \mu_{q(1/\sigma_\nu^2)}\boldsymbol{I}\end{bmatrix}.
\end{aligned}
$$

Combining these expressions leads to the updates in Algorithm 1.

References

BARBER, D. & BISHOP, C.M. (1997). Ensemble learning for multi-layer networks. In *Advances in Neural Information Processing Systems*, vol. 10, eds. M.I. JORDAN, K.J. KEARNS and S.A. SOLLA, pp. 395–401. Cambridge, MA: MIT Press.

CHALLIS, E. & BARBER, D. (2013). Gaussian Kullback–Leibler approximate inference. *J. Mach. Learn. Res.* **14**, 2239–2286.

CRAINICEANU, C.M., RUPPERT, D., CARROLL, R.J., JOSHI, A. & GOODNER, B. (2007). Spatially adaptive Bayesian penalized splines with heteroscedastic errors. *J. Comp. Graph. Stat.* **16**, 265–288.

CRESSIE N.A.C. (1993). *Statistics for Spatial Data*, Revised Edition. New York: Wiley.

FAES, C., ORMEROD, J.T. & WAND, M.P. (2011). Variational Bayesian inference for parametric and nonparametric regression with missing data. *J. Amer. Statist. Assoc.* **106**, 959–971.

HINTON, G.E. & VAN CAMP, D. (1993). Keeping neural networks simple by minimizing the description length of the weights. In *Proceedings of the 6th Annual Association Computing Machinery Conference Computational Learning Theory*, pp. 5–13, New York: The Association for Computing Machinery.

KNOWLES, D.A & MINKA, T.P. (2011). Non-conjugate variational message passing for multinomial and binary regression. In *Advances in Neural Information Processing Systems*, vol. 24, eds. J. SHAWE-TAYLOR, R.S. ZAMEL, P. BARTLETT, F. PEREIRA and K.Q. WEINBERGER, pp. 1701–1709. Cambridge, MA: MIT Press.

LÁZARO-GREDILLA, M. & TITSIAS, M.K. (2011). Variational heteroscedastic Gaussian process regression. *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, Washington, USA.

LUTS, J., BRODERICK, T. & WAND, M.P. (2014). Real-time semiparametric regression. *J. Comp. Graph. Stat.* **23**, 589–615.

LUTS, J., WANG, S.S.J., ORMEROD, J.T. & WAND, M.P. (2015). Semiparametric regression analysis via Infer.NET. Unpublished manuscript.

MAGNUS, J.R. & NEUDECKER, H. (1999). *Matrix Differential Calculus with Applications in Statistics and Econometrics*, Revised Edition. Chichester UK: Wiley.

MARLEY, J.K. & WAND, M.P. (2010). Non-standard semiparametric regression via BRugs. *J. Stat. Softw.* **37**, 1–30.

MENICTAS, M. (2015). Variational inference for heteroscedastic and longitudinal regression models (PhD Thesis). University of Technology, Sydney.

MENICTAS, M. & WAND, M.P. (2013). Variational inference for marginal longitudinal semiparametric regression. *Statistics* **2**, 61–71.

MINKA, T. & WINN, J. (2008). Gates: A graphical notation for mixture models. Microsoft Research Technical Report Series, MSR-TR-2008-185, 1–16.

MINKA, T., WINN, J., GUIVER, J. & KNOWLES, D. (2014). Infer.NET 2.6. Available from URL: http://research.microsoft.com/infernet [Last accessed 26 Feb 2015.]

NOTT, D.J., TRAN, M.-N. & LENG, C. (2012). Variational approximation for heteroscedastic linear models and matching pursuit algorithms. *Stat. Comput.* **22**, 497–512.

OPPER, M. & ARCHAMBEAU, C. (2009). The variational Gaussian approximation revisited. *Neural Comput.* **21**, 786–792.

ORMEROD, J.T. & WAND, M.P. (2010). Explaining variational approximations. *Amer. Statist.* **64**, 140–153.

PEBESMA, E.J. (2004). Multivariate geostatistics in R: The `gstat` package. *Comput. Geosci.* **30**, 683–691.

PEBESMA, E.J. & DUIN, R.N.M. (2005). Spatial patterns of temporal change in North Sea sediment quality on different spatial scales. In *Geostatistics for Environmental Applications: Proceedings of the Fifth European Conference on Geostatistics for Environmental Applications*, eds. P. RENARD, H. DEMOUGEOUT-RENARD and R. FROIDEVAUX, pp. 367–378. Berlin, Heidelberg: Springer-Verlag.

R Core Team (2015). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN: 3-900051-07-0. Available from URL: http://www.R-project.org [Last accessed 26 Feb 2015.]

RAIKO, T., VALPOLA, H., HARVA, M. & KARHUNEN, J. (2007). Building blocks for variational Bayesian learning of latent variable models. *J. Mach. Learn. Res.* **8**, 155–201.

RIGBY, R.A. & STASINOPOULOS, D.M. (2005). Generalized additive models for location, scale and shape. *J. R. Statist. Soc. Ser. C. Appl. Statist.* **54**, 507–554.

RUPPERT, D., WAND, M.P. & CARROLL, R.J. (2003). *Semiparametric Regression.* New York: Cambridge University Press.

RUPPERT, D., WAND, M.P. & CARROLL, R.J. (2009). Semiparametric regression during 2003–2007. *Electron. J. Stat.* **3**, 1193–1256.

SPIEGELHALTER, D.J., THOMAS, A., BEST, N.G., GILKS, W.R. & LUNN, D. (2003). BUGS: Bayesian Inference Using Gibbs Sampling. Medical Research Council Biostatistics Unit, Cambridge, UK. Available from URL: http://www.mrc-bsu.cam.ac.uk/bugs [Last accessed 26 Feb 2015.]

Stan Development Team. (2013). Stan: A C++ library for probability and sampling. Version 1.3. Available from URL: http://mc-stan.org/ [Last accessed 26 Feb 2015.]

WAINWRIGHT, M.J. & JORDAN, M.I. (2008). Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.* **1**, 1–305.

WAND, M.P. (2009). Semiparametric regression and graphical models. *Aust. N. Z. J. Stat.* **51**, 9–41.

WAND, M.P. (2014). Fully simplified Multivariate Normal updates in non-conjugate variational message passing. *J. Mach. Learn. Res.* **15**, 1351–1369.

WAND, M.P. & JONES, M.C. (1995). *Kernel Smoothing.* London: Chapman and Hall.

WAND, M.P. & ORMEROD, J.T. (2008). On semiparametric regression with O'Sullivan penalised splines. *Aust. N. Z. J. Stat.* **50**, 179–198.

WAND, M.P., ORMEROD, J.T., PADOAN, S.A. & FRÜHWIRTH, R. (2011). Mean field variational Bayes for elaborate distributions. *Bayesian Anal.* **6**, 847–900.