Environmental Modelling and Software 125 (2020) 104608

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: http://www.elsevier.com/locate/envsoft

Bayesian spatially varying coefficient models in the spBayes R package

Andrew O. Finley [a,*], Sudipto Banerjee [b]

[a] Department of Forestry, Michigan State University, Natural Resources Building, 480 Wilson Road, Room 126, East Lansing, MI, 48824-6402, USA
[b] Fielding School of Public Health University of California, Los Angeles 650, Charles E. Young Dr. South, Los Angeles, CA, 90095-1772, USA

ARTICLE INFO

Keywords:
MCMC
Multivariate Gaussian process
Kriging
R

ABSTRACT

This paper describes and illustrates new functionality for fitting spatially varying coefficients models in the **spBayes** (version 0.4–2) R package. The new `spSVC` function uses a computationally efficient Markov chain Monte Carlo algorithm and extends current **spBayes** functions, that fit only space-varying intercept regression models, to fit independent or multivariate Gaussian process random effects for any set of columns in the regression design matrix. Newly added OpenMP parallelization options for `spSVC` are discussed and illustrated, as well as helper functions for joint and point-wise prediction and model fit diagnostics. The utility of the proposed models is illustrated using a PM_{10} analysis over central Europe.

1. Introduction

In this paper we describe and illustrate extended functionality of a recent reformulation and rewrite of core functions in the **spBayes** (Finley et al., 2015) R (R Core Team, 2018) package. The **spBayes** package provides a suite of univariate and multivariate regression models for both Gaussian and non-Gaussian outcomes that are spatially indexed. There are, by now, many R packages that provide similar functionality. A recent read of the "Analysis of Spatial Data" CRAN Task View (Bivand, 2019) yielded ~46 packages listed for geostatistical analysis—and this is not an exhaustive accounting of packages available for such analyses. Finley et al. (2015) focused on laying out computationally efficient and flexible MCMC algorithms for estimating an array of spatio-temporal Gaussian process (GP) models. However, while the proposed sampling algorithms were quite general, only a narrow set of models were implemented in **spBayes**. Specifically, users could only specify univariate or multivariate GPs on model intercepts. Now, the addition of the `spSVC` function to **spBayes** (version 0.4–2 available on CRAN July 3, 2019) aims to provide additional user options for placing univariate or multivariate GPs on any set of model regression coefficients.

Such functionality is not unique, there are several R packages capable of fitting spatially varying coefficient (SVC) models. Some are specifically designed to work with spatial or spatio-temporal data and others provide flexibility to allow coefficients to vary by some generic set of variables, which could be indexes in a coordinate system. Most of these packages employ some flavor of spline or kernel based regression method to allow varying impact of predictors. Hastie and Tibshirani (1993) and Fan and Zhang (2008) offer a general development of varying coefficient models and Gelfand et al. (2003) provide treatment particular to spatial settings. Regarding implementation in R, the **spgwr** package (Bivand and Yu, 2017) implements geographically weighted regression as originally detailed in Fotheringham et al. (2002). Key spline-based package options include **mgcv** (Wood, 2017), **svcm** (Heim, 2007), **np** (Hayfield and Racine, 2008), and **mboost** (Hothorn et al., 2018). Bürgin and Ritschard (2017) recently developed a tree-based varying coefficient model (TVCM) algorithm and associated **vcrpart** package. The packages **walker** (Helske, 2019; Vihola et al., 2017), **spTimer** Bakar and Sahu (2018), and **spTDyn** (Bakar et al., 2017, Bakar et al., 2015a, 2015b; 2015b) offer Bayesian time and space-time SVC models. Other Bayesian options include model development using more general software such as INLA (Rue et al., 2009; Lindgren and Rue, 2015; Bakka et al., 2018) and Stan (Carpenter et al., 2017; Stan Development Team, 2018), which can be called from their respective R packages.

The `spSVC` function offers Markov chain Monte Carlo (MCMC) based SVC inference using an efficient sampling algorithm. The algorithm's efficiency derives from updates to only covariance parameters (i.e., regression coefficients and random effects are integrated out), computing parallelization, and use of tuned and/or multi-threaded matrix algebra libraries. Subsequent sections define the model and algorithm specifics, software features, and illustrative analyses of

* Corresponding author.
E-mail addresses: finleya@msu.edu (A.O. Finley), sudipto@ucla.edu (S. Banerjee).
URL: https://www.finley-lab.com (A.O. Finley), https://ph.ucla.edu/faculty/banerjee (S. Banerjee).

https://doi.org/10.1016/j.envsoft.2019.104608
Received 28 October 2019; Accepted 17 December 2019
Available online 9 January 2020
1364-8152/© 2020 Elsevier Ltd. All rights reserved.

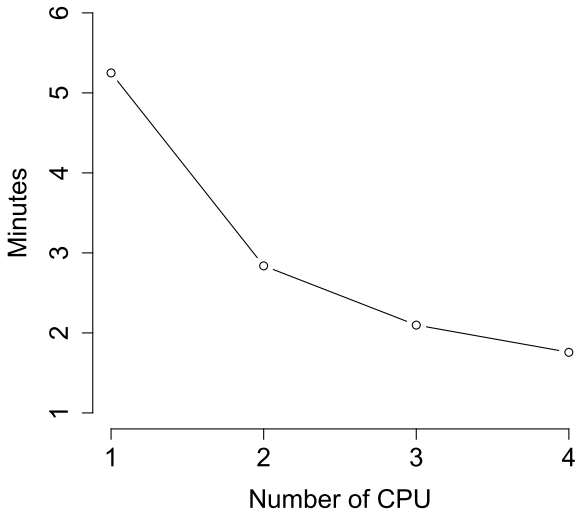


Fig. 1. Simulated data analysis assessment of optimal runtime.

simulated and real data.

2. Models and software

Let $y(\mathbf{s})$ be the dependent variable (response or outcome) at location $\mathbf{s}$ and consider the spatially varying regression model,

$$y(\mathbf{s}) = (\beta_1 + \delta_1 w_1(\mathbf{s})) + \sum_{j=2}^{p} x_j(\mathbf{s})\{\beta_j + \delta_j w_j(\mathbf{s})\} + \varepsilon(\mathbf{s}) , \quad (1)$$

Table 1
Model fit using DIC.

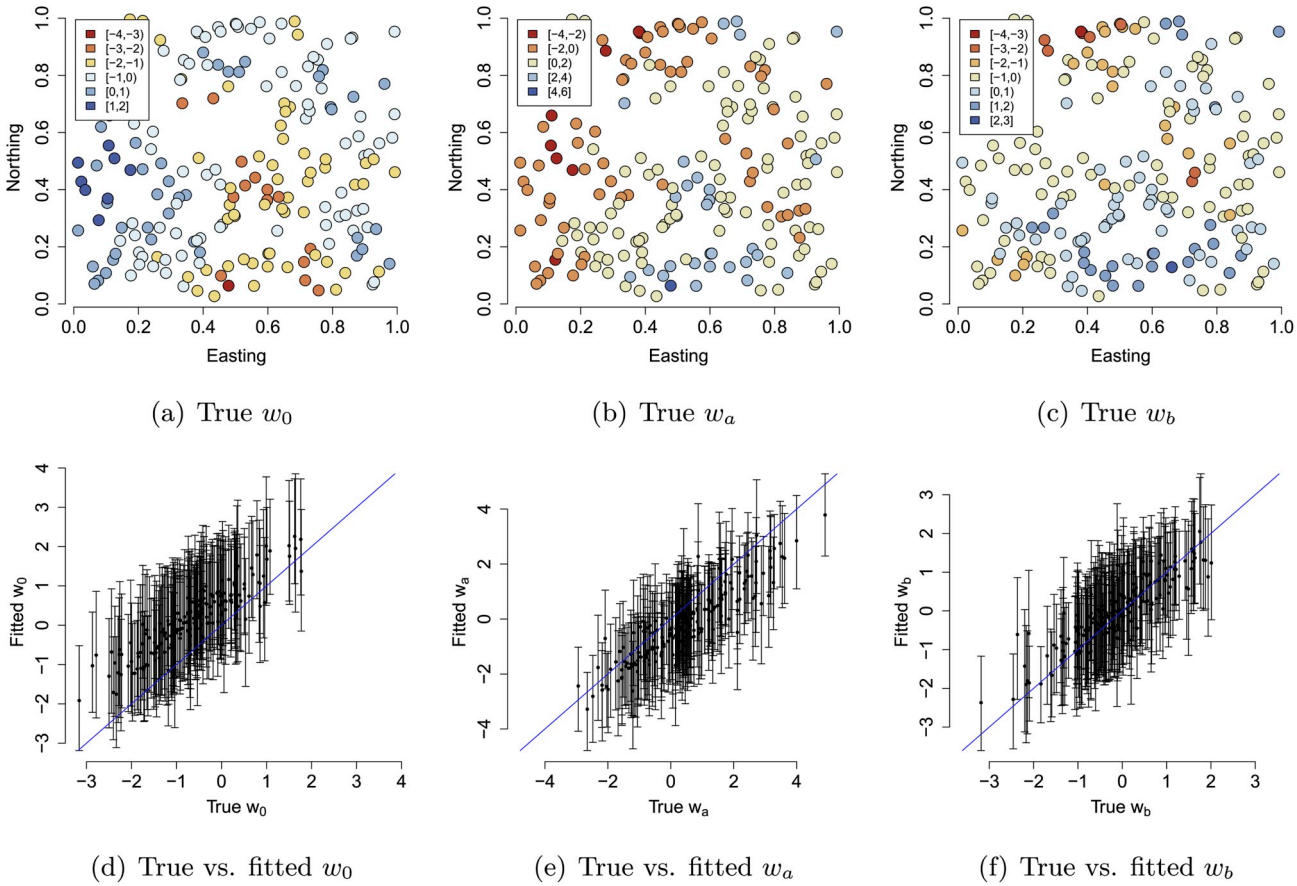
	pD	DIC
Model 1	2.99	363.35
Model 2	84.61	188.88
Model 3	160.53	81.55

where $x_j(\mathbf{s})$, for each $j = 2, \ldots, p$ with $p \geq 1$, is the known value of a predictor at location $\mathbf{s}$, β_j is the regression coefficient corresponding to $x_j(\mathbf{s})$, β_1 is an intercept, and $\varepsilon(\mathbf{s})$ is a Gaussian measurement error process independently distributed for each $\mathbf{s}$. The quantities $w_1(\mathbf{s})$ and $w_j(\mathbf{s})$ are spatial processes corresponding to the intercept and predictors, thereby yielding a spatially varying regression model. We further accommodate the possibility that not all the predictors will have spatially varying impact on the outcome. Thus, δ's in (1) are binary indicators assuming the value 1 if the associated predictor has a spatially varying regression coefficient and 0 otherwise. For later convenience, when the respective $\delta = 1$ we define $\tilde{\beta}_1(\mathbf{s}) = \beta_1 + \delta_1 w_1(\mathbf{s})$ and $\tilde{\beta}_j(\mathbf{s}) = \beta_j + \delta_j w_j(\mathbf{s})$ as the space-varying regression coefficients.

Let $\mathscr{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$ be the set of spatial locations from which $y(\mathbf{s}_i)$ and the predictors have been observed. Let $\mathbf{w}$ be the $nr \times 1$ vector obtained by stacking up $\mathbf{w}(\mathbf{s}_i)$'s, where each $\mathbf{w}(\mathbf{s}_i)$ is an $r \times 1$ vector with j-th

Table 2
Model fit using GPD.

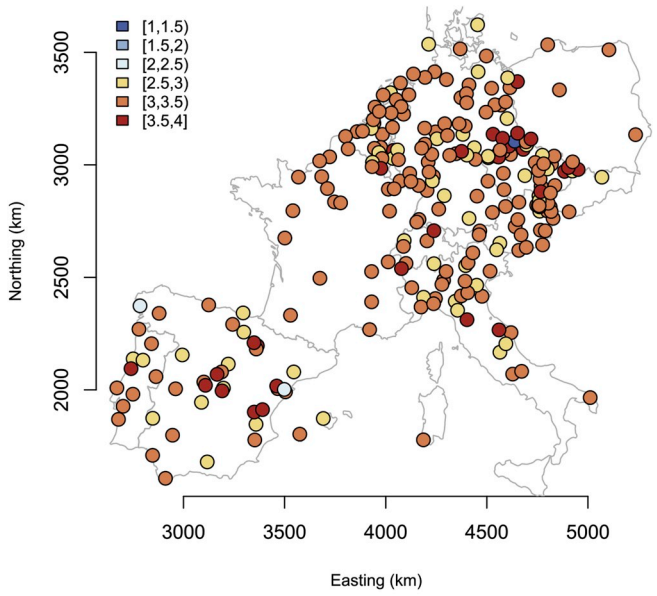
	G	P	D
Model 1	380.53	389.12	769.65
Model 2	94.23	189.33	283.56
Model 3	22.74	122.94	145.68



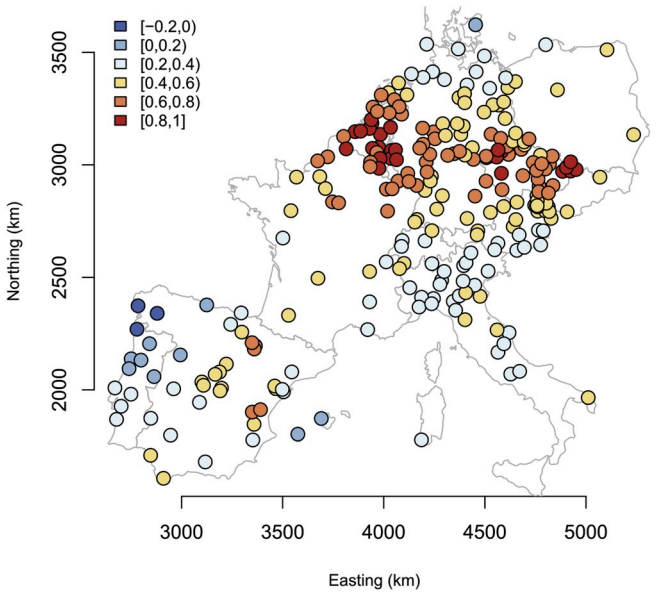
(a) True w_0 (b) True w_a (c) True w_b

(d) True vs. fitted w_0 (e) True vs. fitted w_a (f) True vs. fitted w_b

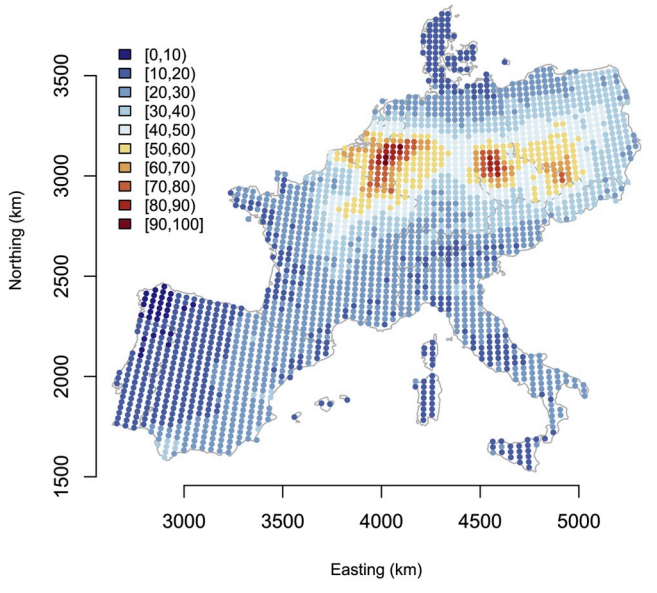
Fig. 2. Simulated data analysis, observed (true) and model estimated (fitted) random effect values. Figures (d)–(f) posterior median and 95% credible interval shown as point and bars, respectively.



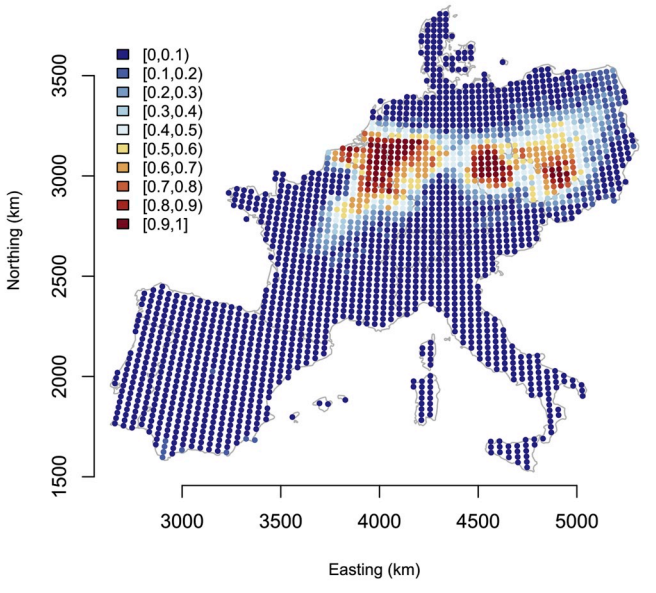
(a) $\tilde{\beta}(\mathbf{s})_0$



(b) $\tilde{\beta}(\mathbf{s})_{CTM}$



(c) Predicted PM_{10}



(d) Probability of $PM_{10} > 50\ \mu g\ m^{-3}$

Fig. 3. Estimated space-varying intercept (a) and CTM regression coefficient (b) over observed monitoring locations. PM_{10} posterior predictive distribution median (c) and probability of regulatory exceedance (d).

entry $w_j(\mathbf{s}_i)$, $j = 1, 2, \ldots, r$ and $i = 1, 2, \ldots, n$. We treat $\mathbf{w}(\mathbf{s})$ as a multivariate Gaussian process (see, e.g., Banerjee et al., 2014) so the matrix $\mathbf{K}_\theta$ is an $nr \times nr$ spatial covariance matrix constructed as a block matrix with (i,j)-th block obtained from the $r \times r$ cross-covariance matrix $\mathbf{K}_\theta(\mathbf{s}_i, \mathbf{s}_j)$ specifying the multivariate spatial process $\mathbf{w}(\mathbf{s})$. In addition, $\boldsymbol{\beta}$ is the $p \times 1$ regression coefficient corresponding to $\mathbf{X}$, and $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$ are the parameters in $\mathbf{K}_\theta$ and $\mathbf{D}_\tau$, respectively.

Consider the Bayesian hierarchical model built from (1),

$$p(\boldsymbol{\tau}, \boldsymbol{\theta}) \times N(\boldsymbol{\beta} \mid \boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta) \times N(\mathbf{w} \mid 0, \mathbf{K}_\theta) \times N(\mathbf{y} \mid \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{w}, \mathbf{D}_\tau) \,, \tag{2}$$

where $\mathbf{y}$ is $n \times 1$ with i-th element $y(\mathbf{s}_i)$, $\mathbf{X}$ is $n \times p$ with the first column $\mathbf{1}$ and the remaining $p-1$ columns corresponding to the predictors $x_j(\mathbf{s}_i)$ in (1). The matrix $\mathbf{Z}$ is $n \times nr$, where $r \leq p$, with precisely those columns of $\mathbf{X}$ which have $\delta_j = 1$.

We will offer users the option to scale and center the matrix $\mathbf{X}$. Note that $\mathbf{Z}$ is constructed from $\mathbf{X}$, and thus, for a scaled and centered $\mathbf{X}$, the predictors used in $\mathbf{Z}$ will also be scaled and centered. Scaling and centering often improves numerical stability and provides more robust estimation of spatially varying regression models (see, e.g., Gelfand et al., 2003).

Some further specifications are in order. In `spSVC` we will fix $\mathbf{D}_\tau = \tau^2\mathbf{I}_n$ so $\boldsymbol{\tau} = \{\tau^2\}$ is the scalar quantity representing the measurement error variance or "nugget" in geostatistics. The cross-covariance matrix $\mathbf{K}_\theta(\mathbf{s}, \mathbf{t})$, where $\mathbf{t}$ is a generic location, will most generally be modeled using the Linear Model of Coregionalization (LMC). Here, we will model $\mathbf{K}_\theta(\mathbf{s}, \mathbf{t}) = \mathbf{A}\Gamma(\mathbf{s}, \mathbf{t})\mathbf{A}^\top$, where $\mathbf{A}$ is an $r \times r$ lower triangular matrix and $\Gamma(\mathbf{s}, \mathbf{t})$ is a diagonal matrix with $\rho_j(\mathbf{s}, \mathbf{t})$ being the j-th diagonal element, where $\rho_j(\mathbf{s}, \mathbf{t})$ is a spatial correlation function with parameters specific to $w_j(\mathbf{s})$. Here, $\boldsymbol{\theta}$ in (2) corresponds to $\{\mathbf{A}, \{\varphi_j\}_{j=1}^r\}$ where each φ_j is a collection of parameters in the spatial correlation function. For example, with the Matérn covariance function each φ_j comprises a spatial decay parameter and a smoothness parameter.

The covariance structure for $\mathbf{w}(\mathbf{s})$ within any location $\mathbf{s}$ is captured by $\mathbf{A}\mathbf{A}^\top$, which identifies with the Cholesky decomposition for $\text{var}\{\mathbf{w}(\mathbf{s})\}$. In general, we will specify priors as

$$p(\tau^2, \boldsymbol{\theta}) = IG(\tau^2 \mid a_\tau, b_\tau) \times IW(\mathbf{A}\mathbf{A}^\top \mid r_a, \mathbf{S}_a) \times \prod_{j=1}^r p(\varphi_j) \,, \tag{3}$$

where *IG* is inverse-Gamma, *IW* is inverse-Wishart, and each $p(\varphi_j)$ can be one of the several distributions provided by **spBayes**. Another particular choice offered by `spSVC` specifies $\mathbf{A} = \text{diag}(\sigma_1, \ldots, \sigma_r)$, so that $\mathbf{K}_\theta(\mathbf{s}, \mathbf{t})$ is diagonal with entries $\sigma_j^2\rho_j(\mathbf{s}, \mathbf{t})$, in which case we assume $IG(\sigma_j^2 \mid a_\sigma, b_\sigma)$ for $j = 1, 2, \ldots, r$. Choices for ρ_j include any of the standard correlation functions offered by **spBayes**.

2.1. Parameter estimation and computational considerations

Bayesian inference for (1) involves sampling the parameters $\boldsymbol{\theta}, \boldsymbol{\beta}$ and $\mathbf{w}$ from their marginal posterior distributions. Such sampling algorithms require expensive operations on dense matrices such as decomposition and multiplication. Therefore, as we have outlined below, care is needed to use efficient numerical algorithms such as Cholesky factorizations, working with triangular systems, and avoiding redundant operations.

2.2. Sampling the process parameters

Sampling from (2) employs MCMC methods, in particular Gibbs sampling and random walk Metropolis steps (e.g., Robert and Casella, 2004). For faster convergence, we integrate out $\boldsymbol{\beta}$ and $\mathbf{w}$ from the model and first sample from $p(\boldsymbol{\theta} \mid \mathbf{y}) \propto p(\boldsymbol{\theta}) \times N(\mathbf{y} \mid \mathbf{X}\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_{y \mid \theta})$, where $\boldsymbol{\Sigma}_{y \mid \theta} = \mathbf{X}\boldsymbol{\Sigma}_\beta\mathbf{X}^\top + \mathbf{Z}\mathbf{K}_\theta\mathbf{Z}^\top + \mathbf{D}_\tau$. This matrix needs to be constructed for every update of $\boldsymbol{\theta}$. $\mathbf{D}_\tau$ is diagonal and $\mathbf{X}\boldsymbol{\Sigma}_\beta\mathbf{X}^\top$ is fixed, so the computation involves the matrix $\mathbf{Z}\mathbf{K}_\theta\mathbf{Z}^\top$ which requires rn^2 flops (floating point operations).

We adopt a random-walk Metropolis step with a multivariate normal proposal (same dimension as there are parameters in $\boldsymbol{\theta}$) after transforming parameters to have support over the entire real line. This involves evaluating

$$\log p(\boldsymbol{\theta} \mid \mathbf{y}) = \text{const} + \log p(\boldsymbol{\theta}) - \frac{1}{2}\log\left|\boldsymbol{\Sigma}_{y \mid \theta}\right| - \frac{1}{2}Q(\boldsymbol{\theta}) \,, \tag{4}$$

where $Q(\boldsymbol{\theta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\mu}_\beta)^\top\boldsymbol{\Sigma}_{y \mid \theta}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\mu}_\beta)$. Generally, we compute $\mathbf{L} = \text{chol}(\boldsymbol{\Sigma}_{y \mid \theta})$, where $chol(\boldsymbol{\Sigma}_{y \mid \theta})$ returns the lower-triangular Cholesky factor $\mathbf{L}$ of $\boldsymbol{\Sigma}_{y \mid \theta}$. This involves $O(n^3/3)$ flops. Next, we obtain $\mathbf{u} = \text{trsolve}(\mathbf{L}, \mathbf{y} - \mathbf{X}\boldsymbol{\mu}_\beta)$, which solves the triangular system $\mathbf{L}\mathbf{u} = \mathbf{y} - \mathbf{X}\boldsymbol{\mu}_\beta$. This involves $O(n^2)$ flops and $Q(\boldsymbol{\theta}) = \mathbf{u}^\top\mathbf{u}$ requires another $2n$ flops. The log-determinant in (4) is evaluated as $2\sum_{i=1}^n \log l_{i,i}$, where $l_{i,i}$ are the diagonal entries in $\mathbf{L}$. Since $\mathbf{L}$ has already been obtained, the log-determinant requires another n steps. Therefore, the Cholesky factorization dominates the work and computing (4) is achieved in $O(n^3)$ flops.

If $\boldsymbol{\beta}$ is flat, i.e., $\boldsymbol{\Sigma}_\beta^{-1} = \mathbf{O}$, the analogue of distribution (4) is

$$\log p(\boldsymbol{\theta} \mid \mathbf{y}) = \text{constant} + \log p(\boldsymbol{\theta}) - \frac{1}{2}\log\left|\mathbf{X}^\top\boldsymbol{\Sigma}_{y \mid \beta,\theta}^{-1}\mathbf{X}\right| - \frac{1}{2}\log\left|\boldsymbol{\Sigma}_{y \mid \beta,\theta}\right| - \frac{1}{2}Q(\boldsymbol{\theta}), \tag{5}$$

where $\boldsymbol{\Sigma}_{y \mid \beta,\theta} = \mathbf{Z}\mathbf{K}_\theta\mathbf{Z}^\top + \mathbf{D}_\tau$ and $Q(\boldsymbol{\theta}) = \mathbf{y}^\top\boldsymbol{\Sigma}_{y \mid \beta,\theta}^{-1}\mathbf{y} - \mathbf{b}^\top(\mathbf{X}^\top\boldsymbol{\Sigma}_{y \mid \beta,\theta}^{-1}\mathbf{X})^{-1}\mathbf{b}$ and $\mathbf{b} = \mathbf{X}^\top\boldsymbol{\Sigma}_{y \mid \beta,\theta}^{-1}\mathbf{y}$. Computations proceed similar to the above. We first evaluate $\mathbf{L} = \text{chol}(\boldsymbol{\Sigma}_{y \mid \beta,\theta})$ and then obtain $[\mathbf{v} : \mathbf{U}] = \text{trsolve}(\mathbf{L}, [\mathbf{y} : \mathbf{X}])$, so $\mathbf{L}\mathbf{v} = \mathbf{y}$ and $\mathbf{L}\mathbf{U} = \mathbf{X}$. Next, we evaluate $\mathbf{W} = \text{chol}(\mathbf{U}^\top\mathbf{U})$, $\mathbf{b} = \mathbf{U}^\top\mathbf{v}$ and solve $\tilde{\mathbf{b}} = \text{trsolve}(\mathbf{W}, \mathbf{b})$. Finally, (5) is evaluated as

$$\log p(\boldsymbol{\theta}) - \sum_{i=1}^p \log w_{i,i} - \sum_{i=1}^n \log l_{i,i} - \frac{1}{2}(\mathbf{v}^\top\mathbf{v} - \tilde{\mathbf{b}}^\top\tilde{\mathbf{b}}),$$

where $w_{i,i}$'s and $l_{i,i}$'s are the diagonal elements in $\mathbf{W}$ and $\mathbf{L}$ respectively. The number of flops is again of cubic order in n.

Importantly, our strategy above avoids computing inverses. We use Cholesky factorizations and solve only triangular systems. If n is not large, say $\sim 10^2$, this strategy is feasible. As described in Section 2.2 and illustrated in Section 3, use of efficient and parallelized numerical linear algebra routines yields substantial gains in computing time.

2.3. Sampling the slope and the random effects

Once we have obtained marginal posterior samples $\boldsymbol{\theta}$ from $p(\boldsymbol{\theta} \mid \mathbf{y})$, we can draw posterior samples of $\boldsymbol{\beta}$ and $\mathbf{w}$ using *composition sampling*. Suppose $\{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(M)}\}$ are M samples from $p(\boldsymbol{\theta} \mid \mathbf{y})$. Drawing $\boldsymbol{\beta}^{(k)} \sim p(\boldsymbol{\beta} \mid \boldsymbol{\theta}^{(k)}, \mathbf{y})$ and $\mathbf{w}^{(k)} \sim p(\mathbf{w} \mid \boldsymbol{\theta}^{(k)}, \mathbf{y})$ for $k = 1, 2, \ldots M$ results in M samples from $p(\boldsymbol{\beta} \mid \mathbf{y})$ and $p(\mathbf{w} \mid \mathbf{y})$ respectively. Only the samples of $\boldsymbol{\theta}$ obtained after convergence (i.e., *post burn-in*) of the MCMC algorithm need to be stored.

To elucidate further, note that $\boldsymbol{\beta} \mid \boldsymbol{\theta}, \mathbf{y} \sim N_p(\mathbf{B}\mathbf{b}, \mathbf{B})$ with mean $\mathbf{B}\mathbf{b}$ and variance-covariance matrix $\mathbf{B}$, where

$$\mathrm{b} = \boldsymbol{\Sigma}_\beta^{-1}\boldsymbol{\mu}_\beta + \mathbf{X}^\top\boldsymbol{\Sigma}_{y \mid \beta,\theta}^{-1}\mathbf{y} \quad \text{and} \quad \mathbf{B} = \left(\boldsymbol{\Sigma}_\beta^{-1} + \mathbf{X}^\top\boldsymbol{\Sigma}_{y \mid \beta,\theta}^{-1}\mathbf{X}\right)^{-1} . \tag{6}$$

For each $k = 1, 2, \ldots, M$, we compute $\mathbf{B}$ and $\mathbf{b}$ at the current value $\boldsymbol{\theta}^{(k)}$ and draw $\boldsymbol{\beta}^{(k)} \sim N_p(\mathbf{B}\mathbf{b}, \mathbf{B})$. This is achieved by computing $\mathbf{b} = \boldsymbol{\Sigma}_\beta^{-1}\boldsymbol{\mu}_\beta + \mathbf{U}^\top\mathbf{v}$, where $\mathbf{L} = chol(\boldsymbol{\Sigma}_{y \mid \beta,\theta^{(k)}})$ and $[\mathbf{v} : \mathbf{U}] = \text{trsolve}(\mathbf{L}, [\mathbf{y} : \mathbf{X}])$. Next, we generate p independent standard normal variables, collect them into $\mathbf{z}$ and set

$$\boldsymbol{\beta}^{(k)} = \text{trsolve}\left(\mathbf{L}_B^\top, \text{trsolve}(\mathbf{L}_B, \mathbf{b})\right) + \text{trsolve}\left(\mathbf{L}_B^\top, \mathbf{z}\right) , \tag{7}$$

where $\mathbf{L}_B = \text{chol}(\boldsymbol{\Sigma}_\beta^{-1} + \mathbf{U}^\top\mathbf{U})$. This completes the k-th iteration. After M iterations, we obtain $\{\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\ldots,\boldsymbol{\beta}^{(M)}\}$, which are samples from $p(\boldsymbol{\beta} \mid \mathbf{y})$.

Mapping point or interval estimates of spatial random effects is often helpful in identifying missing regressors and/or building a better understanding of model adequacy. $\boldsymbol{\Sigma}_{y \mid w,\theta} = \mathbf{X}\boldsymbol{\Sigma}_\beta\mathbf{X}^\top + \mathbf{D}_\tau$ and note that $\mathbf{w} \mid \boldsymbol{\theta},\mathbf{y} \sim N(\mathbf{Bb},\mathbf{B})$, where

$$\mathrm{b} = \mathbf{Z}^\top\boldsymbol{\Sigma}_{y \mid w,\theta}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\mu}_\beta) \quad \text{and} \quad \mathbf{B} = \left(\mathbf{K}_\theta^{-1} + \mathbf{Z}^\top\boldsymbol{\Sigma}_{y \mid w,\theta}^{-1}\mathbf{Z}\right)^{-1}. \tag{8}$$

The vector $\mathbf{b}$ here is computed analogously as for $\boldsymbol{\beta}$. For each $k = 1, 2, \ldots, M$ we now evaluate $\mathbf{L} = \text{chol}(\boldsymbol{\Sigma}_{y \mid w,\theta^{(k)}})$, $[\mathbf{v} : \mathbf{U}] = \text{trsolve}(\mathbf{L}, [\mathbf{y} - \mathbf{X}\boldsymbol{\mu}_\beta : \mathbf{Z}(\boldsymbol{\theta}^{(k)})])$ and set $\mathbf{b} = \mathbf{U}(\boldsymbol{\theta}^{(k)})^\top\mathbf{v}$. For computing $\mathbf{B}$, one could proceed as for $\boldsymbol{\beta}$ but that would involve $\text{chol}(\mathbf{K}(\boldsymbol{\theta}))$, which may become numerically unstable for certain covariance functions (e.g., the Gaussian or the Matérn with large ν). For robust software performance we define $\mathbf{G}_\theta^{-1} = \mathbf{Z}'\boldsymbol{\Sigma}_{y \mid w,\theta}^{-1}\mathbf{Z}$ and utilize the identity (Henderson and Searle, 1981)

$$\left(\mathbf{K}_\theta^{-1} + \mathbf{G}_\theta^{-1}\right)^{-1} = \mathbf{G}_\theta - \mathbf{G}_\theta(\mathbf{K}_\theta + \mathbf{G}_\theta)^{-1}\mathbf{G}_\theta$$

to devise a numerically stable algorithm. For each $k = 1, 2, \ldots, M$, we evaluate $\mathbf{L} = \text{chol}(\mathbf{K}_\theta^{(k)} + \mathbf{G}_\theta^{(k)})$, $\mathbf{W} = \text{trsolve}(\mathbf{L}, \mathbf{G}_\theta^{(k)})$ and $\mathbf{L}_B = \text{chol}(\mathbf{G}_\theta^{(k)} - \mathbf{W}^\top\mathbf{W})$. If $\mathbf{z}$ is a $r \times 1$ vector of independent standard normal variables, then we set $\mathbf{w}^{(k)} = \mathbf{L}_B\mathbf{L}_B^\top\mathbf{b} + \mathbf{L}_B\mathbf{z}$. The resulting $\{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(M)}\}$ are samples from $p(\mathbf{w} \mid \mathbf{y})$.

2.4. Spatial predictions

To predict a random $n_0 \times 1$ vector $\mathbf{y}_0$ associated with a $n_0 \times p$ matrix of predictors, $\mathbf{X}_0$, we assume that

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_0 \end{bmatrix} \Big| \boldsymbol{\beta},\boldsymbol{\theta} \sim N_{n_0+n}\left(\begin{bmatrix} \mathbf{X} \\ \mathbf{X}_0 \end{bmatrix}\boldsymbol{\beta}, \begin{bmatrix} \mathbf{C}_{11}(\boldsymbol{\theta}) & \mathbf{C}_{12}(\boldsymbol{\theta}) \\ \mathbf{C}_{12}(\boldsymbol{\theta})^\top & \mathbf{C}_{22}(\boldsymbol{\theta}) \end{bmatrix}\right), \tag{9}$$

where $\mathbf{C}_{11}(\boldsymbol{\theta}) = \boldsymbol{\Sigma}_{y \mid \beta,\theta}$, $\mathbf{C}_{12}(\boldsymbol{\theta})$ is the $n \times n_0$ cross-covariance matrix between $\mathbf{y}$ and $\mathbf{y}_0$, and $\mathbf{C}_{22}(\boldsymbol{\theta})$ is the variance-covariance matrix for $\mathbf{y}_0$. A valid *joint* distribution will supply a conditional distribution $p(\mathbf{y}_0 \mid \mathbf{y},\boldsymbol{\beta},\boldsymbol{\theta})$, which is normal with mean and variance

$$\boldsymbol{\mu}_p = \mathbf{X}_0\boldsymbol{\beta} + \mathbf{C}_{12}(\boldsymbol{\theta})^\top\mathbf{C}_{11}(\boldsymbol{\theta})^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \text{ and } \boldsymbol{\Sigma}_p = \mathbf{C}_{22}(\boldsymbol{\theta}) - \mathbf{C}_{12}(\boldsymbol{\theta})^\top\mathbf{C}_{11}(\boldsymbol{\theta})^{-1}\mathbf{C}_{12}(\boldsymbol{\theta}) \tag{10}$$

Bayesian prediction proceeds by sampling from the posterior predictive distribution $p(\mathbf{y}_0 \mid \mathbf{y}) = \int p(\mathbf{y}_0 \mid \mathbf{y},\boldsymbol{\beta},\boldsymbol{\theta})p(\boldsymbol{\beta},\boldsymbol{\theta} \mid \mathbf{y})d\boldsymbol{\beta}d\boldsymbol{\theta}$. For each posterior sample of $\{\boldsymbol{\beta},\boldsymbol{\theta}\}$, we draw a corresponding $\mathbf{y}_0 \sim N(\boldsymbol{\mu}_p,\boldsymbol{\Sigma}_p)$. This produces samples from the posterior predictive distribution.

The posterior predictive computations involve only the retained MCMC samples after convergence. For any posterior sample $\{\boldsymbol{\beta}^{(k)},\boldsymbol{\theta}^{(k)}\}$, we solve $[\mathbf{u} : \mathbf{V}] = \text{trsolve}(\mathbf{L}, [\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^{(k)} : \mathbf{C}_{12}(\boldsymbol{\theta}^{(k)})])$, where $\mathbf{L} = \text{chol}(\mathbf{C}_{11}(\boldsymbol{\theta}^{(k)}))$. Next, we set $\boldsymbol{\mu}_p^{(k)} = \mathbf{X}_0\boldsymbol{\beta}^{(k)} + \mathbf{V}^\top\mathbf{u}$ and $\boldsymbol{\Sigma}_p^{(k)} = \mathbf{C}_{22}(\boldsymbol{\theta}^{(k)}) - \mathbf{V}^\top\mathbf{V}$ and draw $\mathbf{y}_0^{(k)} \sim N(\boldsymbol{\mu}_p^{(k)},\boldsymbol{\Sigma}_p^{(k)})$.

Updating $\mathbf{y}_0^{(k)}$'s requires Cholesky factorization of $\boldsymbol{\Sigma}_p$, which is $n_0 \times n_0$ and can be expensive if n_0 is large. In most practical settings, it is sufficient to take $n_0 = 1$ and perform *point-wise* predictions.

2.5. Software features

The `spSVC` function accommodates the `spLM` function in **spBayes** and offers additional user options to simplify analysis and inference. The list below highlights some of these new options.

1. Any set of predictors can receive either independent univariate GPs or a multivariate GP.
2. Prediction can be done by sampling from either the joint or point-wise (marginal) posterior predictive distribution.
3. OpenMP(Dagum and Menon, 1998) support is available via the `n.omp.threads` argument for parameter estimation, composition sampling, model fit diagnostics, and prediction functions.
4. Matrix operation parallelization is available via multi-threaded implementations of Basic Linear Algebra Subprograms (BLAS; www.netlib.org/blas) and `Linear Algebra Package (LAPACK;` www.netlib.org/lapack).
5. Coordinate system used to index observed and prediction locations can be of arbitrary dimension—users were previously restricted to using 2-dimensional systems.
6. Univariate and multivariate random effect samples and space-varying coefficients are returned as lists with element names corresponding to the given predictor.

3. Illustrations

We consider two analyses to illustrate key features of `spSVC` along with supporting functions. The first analysis is of a simulated dataset and second is of an air pollution dataset that was previously analyzed in Hamm et al. (2015) and Datta et al. (2016).

3.1. Analysis of simulated data

The simulated data `mvSVCData` is available in **spBayes** and comprises $n = 500$ observations distributed within a 2-dimensional unit square spatial domain. At generic location $\mathbf{s}$ the outcome was generated following

$$y(\mathbf{s}) = \beta_0 + w_0(\mathbf{s}) + a(\mathbf{s})\{\beta_a + w_a(\mathbf{s})\} + b(\mathbf{s})\{\beta_b + w_b(\mathbf{s})\} + \varepsilon(\mathbf{s}). \tag{11}$$

The predictors $a(\mathbf{s})$ and $b(\mathbf{s})$ were drawn from independent normal distributions with mean zero and variance one. The regression coefficients β_0, β_a, and β_b equaled 1, 10, and -10, respectively. The $r = 3$ spatial random effects associated with the intercept and predictors were generated from a non-separable multivariate GP. The cross-covariance function used to construct the (i,j)-th $r \times r$ block in the multivariate GP's $nr \times nr$ covariance matrix, i.e., $\mathbf{K}_\theta(\mathbf{s}_i,\mathbf{s}_j) = \mathbf{A}\Gamma(\mathbf{s}_i,\mathbf{s}_j)\mathbf{A}^\top$, was

$$\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 1 & 0.1 \end{pmatrix}\begin{pmatrix} \exp(-4d_{i,j}) & 0 & 0 \\ 0 & \exp(-6d_{i,j}) & 0 \\ 0 & 0 & \exp(-6d_{i,j}) \end{pmatrix}\begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0.1 \end{pmatrix}, \tag{12}$$

where $d_{i,j}$ is the euclidean distance between location $\mathbf{s}_i$ and $\mathbf{s}_j$, and diagonal elements of $\Gamma(\mathbf{s}_i,\mathbf{s}_j)$ are the exponential correlation function $\exp(-\varphi_k d_{i,j})$ for $k = 1, 2, \ldots, r$. Fig. 2(a) display the realizations of $\mathbf{w}_0$, $\mathbf{w}_a$, and $\mathbf{w}_b$. The residual term $\varepsilon(\mathbf{s})$ was simulated from a Normal distribution with mean zero and variance $\tau^2 = 0.1$.

The code below specifies the model covariance parameters' prior distributions, and MCMC sampler starting and Metropolis proposal variance values. Here, we use a Uniform prior for the spatial decay parameters each with support from 1 to 10. The prior for the cross-covaraince matrix is an IW with degrees of freedom r and identity scale matrix. The prior for the measurement error (or nugget variance) follows an IG with shape 2 and scale 1.

The parameter priors, starting values, and Metropolis sampler proposal variances are passed to `spSVC` via the `priors`, `starting`, and `tuning` arguments, respectively. The proposed model is specified via the `formula` argument using syntax like that used in base R's `lm`, with the addition of the `svc.cols` argument that accepts a vector of either integer indexes or character names to indicate the space-varying design matrix columns (i.e., the columns of $\mathbf{X}$ with $\delta_j = 1$). For example, in the call to `spSVC` below, the vector passed to `svc.cols` indicates we want the intercept and columns labeled `a` and `b` to follow a multivariate GP (or, equivalently, one could use the argument value `c(1,2,3)`).

```
data(SVCMvData.dat)

r <- 3

n.ltr <- r*(r+1)/2

priors <- list("phi.Unif"=list(rep(1,r), rep(10,r)),
               "K.IW"=list(r, diag(rep(1,r))),
               "tau.sq.IG"=c(2, 1))

starting <- list("phi"=rep(3/0.5,r), "A"=c(1,0,0,1,0,1), "tau.sq"=1)

tuning <- list("phi"=rep(0.1,r), "A"=rep(0.01, n.ltr), "tau.sq"=0.01)

sim.m <- spSVC(y~a+b, coords=c("x.coords","y.coords"), data=SVCMvData.dat,
             starting=starting, svc.cols=c("(Intercept)","a","b"),
             tuning=tuning, priors=priors, cov.model="exponential",
             n.samples=10000, n.report=5000, n.omp.threads=4)
----------------------------------------
General model description
----------------------------------------
Model fit with 200 observations.

Number of covariates 3.

Number of space varying covariates 3.

Using the exponential spatial correlation model.

Number of MCMC samples 10000.

Priors and hyperpriors:
beta flat.
K IW hyperpriors:
df: 3.00000
S:
1.000 0.000 0.000
0.000 1.000 0.000
0.000 0.000 1.000

phi Unif lower bound hyperpriors: 1.000 1.000 1.000
phi Unif upper bound hyperpriors: 10.000 10.000 10.000

tau.sq IG hyperpriors shape=2.00000 and scale=1.00000

Source compiled with OpenMP, posterior sampling is using 4 thread(s).
------------------------------------------------
Sampling
------------------------------------------------
Sampled: 5000 of 10000, 50.00%
Report interval Metrop. Acceptance rate: 34.64%
Overall Metrop. Acceptance rate: 34.64%
------------------------------------------------
Sampled: 10000 of 10000, 100.00%
Report interval Metrop. Acceptance rate: 34.24%
Overall Metrop. Acceptance rate: 34.44%
------------------------------------------------
```

As described in Section 2, spSVC computes and returns MCMC samples for only model covariance parameters. If verbose = TRUE, basic model specifications are written to the terminal followed by updates on the sampler's progress and Metropolis algorithm acceptance rate. The sampler progress report interval is controlled using the n.report argument. One should adjust the Metropolis sampler proposal variances to achieve an acceptance rate between ∼30-50% (see, e.g., Gelman et al., 2013, for model fitting best practices). If it proves difficult to maintain an acceptable acceptance rate, the amcmc argument can be added to invoke an adaptive MCMC algorithm (Roberts and Rosenthal, 2009) that automatically adjusts the tuning to achieve a target acceptance rate (see the manual page for more details).

The n.omp.threads argument in spSVC call above requests that key for loops within a given MCMC iteration use 4 threads via openMP (Dagum and Menon, 1998). If the user's R is set up to use a parallelized version of BLAS then n.omp.threads will also control the number of threads in some LAPACK matrix operations. Such parallelization can greatly reduce the sampler's runtime.

The computer used to conduct this analysis has an Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz chip with 4 cores and R compiled with openMP, as confirmed in the "General model description" printed after calling spSVC, which notes Source compiled with OpenMP, posterior sampling is using 4 thread(s). spSVC will throw a warning if R was not compiled with openMP support and n.omp.threads is set to a value greater than 1. In addition to openMP support, the current implementation of R uses openBLAS (Zhang, 2016) which is a version of BLAS capable of exploiting multiple processors. Fig. 1 shows the runtime needed to complete 10000 MCMC iterations across the number of available CPUs.

Following execution of spSVC, the sim.m object holds MCMC samples for covariance parameters (p.theta.samples) along with data and model fitting details. Using, possibly post burn-in and thinned, p.theta.samples, the spRecover function conducts composition sampling to generate samples from the regression coefficients $\boldsymbol{\beta}$ (p.beta.recover.samples), spatial random effects $\mathbf{w}$ (p.w.recover.samples), and model fitted values (p.y.samples). spRecover also returns the subset of p.theta.samples (p.theta.recover.samples) used in the composition sampling. Further, for convenience, spRecover returns samples of the space-varying regression coefficients $\tilde{\beta}(\mathbf{s})$'s. spRecover appends these various composition sampling outputs to the spSVC input object, i.e., the sim.m object returned by spRecover below is identical to the sim.m object returned by spSVC except for the addition of the composition sampling results. In addition to providing posterior samples for all model parameters, a call to spRecover is necessary for subsequent prediction and model fit diagnostics, via spPredict and spDiag respectively. Like spSVC, spRecover takes advantage of multiple CPUs via openMP when available.

```
sim.m <- spRecover(sim.m, start=5000, thin=2, n.omp.threads=4, verbose=FALSE)
```

spSVC and spRecover return samples as coda objects to simplify posterior summaries. Output below provides the post burn-in and thinned median with lower and upper 95% credible bounds for β's, cross-covariance matrix used to construct $\mathbf{K}_\theta$, spatial decays φ's, and τ^2. These summaries show the model captures well the parameter values used to simulate the data.

```
round(summary(sim.m$p.beta.recover.samples)$quantiles[,c(3,1,5)],2)

                50%   2.5% 97.5%
(Intercept)    0.20  -1.05  1.16
a             10.53   9.37 11.92
b            -10.20 -11.08 -9.31
```

Note, following the notation in Section 2 the cross-covaraince matrix used to simulated the data is

$$\mathbf{K}_\theta = \mathbf{A}\mathbf{A}^\top \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 1 & 0.1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0.1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 1 \\ 0 & 1 & 1.01 \end{pmatrix}. \quad (13)$$

The posterior summary of the covariance parameters is below. Observed versus estimated random effects are given Fig. 2(d)–(f).

```
round(summary(sim.m$p.theta.recover.samples)$quantiles[,c(3,1,5)],2)

                   50%  2.5% 97.5%
K[1,1]            1.11  0.58  2.78
K[2,1]           -1.00 -2.49 -0.46
K[3,1]           -0.06 -0.52  0.35
K[2,2]            1.89  1.17  3.60
K[3,2]            0.90  0.35  1.76
K[3,3]            1.14  0.63  2.03
tau.sq            0.18  0.10  0.33
phi.(Intercept)   4.16  1.57  8.65
phi.a             4.98  2.09  9.81
phi.b             6.13  1.55  9.89
```

3.2. Analysis of air pollution data

Increases in human morbidity and mortality is a known outcome to airborne particulate matter (PM) exposure (Brunekreef and Holgate, 2002; Loomis et al., 2013; Hoek et al., 2013). In response, regulatory agencies have instigated monitor programs and regulate PM concentrations. One such regulation by the European Commission's air quality standards limits PM_{10} (PM<10 μm in diameter) concentrations to 50 μg m^{-3} average over 24 h and 40 μg m^{-3} over a year (European Commission, 2015).

Measurements made with instruments at monitoring stations are considered authoritative; however, these observations are often too sparse to deliver regional maps at sufficient resolution to assess progress with mitigation strategies and for monitoring compliance. One solution is to couple spatially sparse monitoring station observations with spatially complete chemistry transport model (CTM) output, (see, e.g., van de Kassteele and Stein, 2006; Denby et al., 2008; Candiani et al., 2013). In such settings, monitoring station observations serve as a regression model outcome with CTM output set as a predictor.

$$PM_{10}(\mathbf{s}) = \beta_0 + w_0(\mathbf{s}) + \text{CTM}(\mathbf{s})\{\beta_{CTM} + w_{CTM}(\mathbf{s})\} + \varepsilon(\mathbf{s}) \,. \quad (14)$$

This illustration draws on data and analyses presented in (Hamm et al., 2015; Datta et al., 2016). We consider April 6, 2010, PM_{10} measurements across central Europe with corresponding output from the LOTOS-EUROS (Schaap et al., 2008) CTM. Following (Hamm et al., 2015) we hypothesis a space-varying relationship between the PM_{10} measurements observed at monitoring stations and CTM output. In what follows, we compare fit metrics for three candidate models derived from (14): 1) a non-spatial regression; 2) space-varying intercept; 3) space-varying intercept and CTM output. Resulting model objects are called `pm.1`, `pm.2`, and `pm.3`, respectively. For brevity, code only for fitting `pm.3` is shown. We then consider parameter estimates and associated plots of the spatial random effects from `pm.3`, followed by development of predictive maps of both the space-varying coefficients and PM_{10} prediction for a grid over the study area.

We begin by loading the data and separating it into a "model" set `PM10.mod` comprising locations where both PM_{10} measurements and CTM values are available, and a "prediction" set `PM10.pred` where only CTM values are available. Here too, we calculated the maximum distance between any two monitoring stations which will help with setting prior distributions for spatial decay parameters.

```
data(PM10.dat)

PM10.mod <- PM10.dat[!is.na(PM10.dat$pm10.obs),]
PM10.pred <- PM10.dat[is.na(PM10.dat$pm10.obs),]

d.max <- max(iDist(PM10.mod[,c("x.coord","y.coord")]))
d.max #km

[1] 2929.193
```

The code below specifies the model covariance parameters' prior distributions, and MCMC sampler starting and Metropolis proposal variance values. Unlike the simulated data analysis, here we demonstrate placing independent GPs on the intercept and CTM predictor. This requires priors for a process specific spatial decay parameter φ and variance σ^2. We again use a Uniform prior for the process' decay parameters that provides support for an effective spatial range between ~ 3 and 2197 km, given an exponential covariance function. The two spatial variances and single observational variance τ^2 each are assumed to follow an IG with shape 2 and scale 1. We center the IG's on 1, because it is approximately equal to the residual variance from the first candidate model, i.e., the non-spatial regression. One should generally do careful exploratory data analysis to arrive at a robust set of prior distributions and hyperparameters.

```
r <- 2

priors <- list("phi.Unif"=list(rep(3/(0.75*d.max), r), rep(3/(0.001*d.max), r)),
               "sigma.sq.IG"=list(rep(2, r), rep(1, r)),
               "tau.sq.IG"=c(2, 1))

starting <- list("phi"=rep(3/(0.1*d.max), r), "sigma.sq"=rep(1, r), "tau.sq"=1)

tuning <- list("phi"=rep(0.1, r), "sigma.sq"=rep(0.05, r), "tau.sq"=0.1)

n.samples <- 10000

m.3 <- spSVC(pm10.obs ~ pm10.ctm, coords=c("x.coord","y.coord"),
             data=PM10.mod, starting=starting, svc.cols=c(1,2),
             tuning=tuning, priors=priors, cov.model="exponential",
             n.samples=n.samples, n.report=5000, n.omp.threads=4)

----------------------------------------
General model description
----------------------------------------
Model fit with 256 observations.

Number of covariates 2.

Number of space varying covariates 2.

Using the exponential spatial correlation model.

Number of MCMC samples 10000.

Priors and hyperpriors:
beta flat.
Diag(K) IG hyperpriors
parameter shape scale
K[1,1] 2.000000 1.000000
K[2,2] 2.000000 1.000000

phi Unif lower bound hyperpriors: 0.001 0.001
phi Unif upper bound hyperpriors: 1.024 1.024

tau.sq IG hyperpriors shape=2.00000 and scale=1.00000

Source compiled with OpenMP, posterior sampling is using 4 thread(s).
------------------------------------------------
Sampling
------------------------------------------------
Sampled: 5000 of 10000, 50.00%
Report interval Metrop. Acceptance rate: 36.84%
Overall Metrop. Acceptance rate: 36.84%
------------------------------------------------
Sampled: 10000 of 10000, 100.00%
Report interval Metrop. Acceptance rate: 36.08%

Overall Metrop. Acceptance rate: 36.46%
------------------------------------------------
```

We again pass the `spSVC` object to `spRecover` for composition sampling of the remaining model parameters needed for posterior summaries, model assessment, and subsequent prediction.

```
m.3 <- spRecover(m.3, start=floor(0.75*n.samples), thin=2,
                 n.omp.threads=4, verbose=FALSE)
```

Passing the `spRecover` object to `spDiag` yields several popular model fit diagnostics, two of which are summarized in Tables 1 and 2. Table 1 shows the deviance information criterion (DIC) and associated effective number of parameters pD (Spiegelhalter et al., 2001), while Table 2 presents a posterior predictive loss metric $D = G + P$ proposed by (Gelfand and Ghosh, 1998), where G measures goodness of fit and P penalizes complexity. Models with lower values of DIC or D are preferred over those with higher values. Both metrics favor Model 3 which allows both the intercept and CTM predictor to vary spatially over the study area.

Again, passing `spRecover`'s `coda` objects to `summary` provides posterior summaries of regression coefficients and covariance parameters.

```
round(summary(m.3$p.beta.recover.samples)$quantiles[,c(3,1,5)],3)

              50%   2.5% 97.5%
(Intercept) 3.189  2.105 4.286
pm10.ctm    0.324 -0.087 0.726

round(summary(m.3$p.theta.recover.samples)$quantiles[,c(3,1,5)],3)

                      50%  2.5% 97.5%
sigma.sq.(Intercept) 0.278 0.146 0.480
sigma.sq.pm10.ctm    0.103 0.066 0.153
tau.sq               0.286 0.137 0.463
phi.(Intercept)      0.426 0.072 0.909
phi.pm10.ctm         0.001 0.001 0.002
```

Given the spatial decay parameter estimates, the corresponding effective spatial range (defined as the distance at which the correlation drops to 0.05) posterior median and 95% CI for the intercept and CTM processes are approximately 7.03 (3.3, 41.82) km and 2005.4 (1330, 2183.12) km, respectively. While the CTM predictor does have a long spatial range relative to the size of the study area, model fit metrics and the magnitude of its process variance `sigma.sq.pm10.ctm` estimates relative to the intercept process and nugget variance, offer evidence for a space-varying relationship with the outcome variable. This conclusion is further reinforced by Fig. 3(b), which shows the posterior median for the CTM predictor regression coefficient, $\tilde{\beta}_{CTM}(\mathbf{s})$'s, over observed monitoring locations. These posterior samples, along with those of the space-varying intercept, $\tilde{\beta}_0(\mathbf{s})$'s, are extracted from `m.3` and summarized in the code below (`tilde.beta.0` and `tilde.beta.ctm` are displayed in Fig. 3(a)–(b)).

```
tilde.beta.0 <- apply(
    m.3$p.tilde.beta.recover.samples[["tilde.beta.(Intercept)"]],
    1, median)

tilde.beta.ctm <- apply(
    m.3$p.tilde.beta.recover.samples[["tilde.beta.pm10.ctm"]],
    1, median)
```

We next turn to prediction over the grid of 2336 CTM output locations via a call to `spPredict`. As illustrated below, this call uses samples from a `spRecover` object along with the prediction locations (`pred.coords`) and associated design matrix (`pred.covars`). The argument `joint` specifies if posterior predictive samples should be drawn from the joint or point-wise distribution.

```
m.3.pred <- spPredict(m.3, pred.covars=cbind(1, PM10.pred$pm10.ctm),
                      pred.coords=PM10.pred[,1:2], thin=25,
                      joint=TRUE, n.omp.threads=4, verbose=FALSE)
```

If the number of prediction locations is large, joint prediction can be prohibitively expensive. Even here with 2336 locations, 51 samples, and using 4 CPUs, joint posterior sampling takes 3.53 min verses 0.72 min for point-wise sampling.

Joint prediction results are given in the bottom row of Fig. 3. The posterior predictive distribution median map (Fig. 3(c)) shows three distinct zones of high PM_{10} values over Central Europe. A compelling quality of MCMC-based inference is access to the posterior predictive distribution. This access facilitates summaries like that given in Fig. 3(d) which identifies the probability that a given location will exceed a PM_{10} value 50 μg m^{-3} (as further explored in Hamm et al. (2015) and Datta et al. (2016)).

4. Summary

The new `spSVC` function more fully implements the computationally efficient MCMC algorithm detailed in Finley et al. (2015) and provides a flexible software tool for fitting spatially varying coefficient models. While other software, some of which are noted in Section 1, offer similar spatially adaptive regression, few provide both univariate and multivariate GP specifications and the computational efficiency delivered by the proposed sampling algorithm and use of OpenMP parallelization in combination with optional calls to multi-lower-level BLAS and LAPACK multi-threaded matrix algebra libraries. Future work will focus on extending this function to accommodate non-Guassian and multivariate outcomes, as well as for settings where the number of locations precluded the use of full-rank spatial GPs.

Acknowledgments

Finley was supported by National Science Foundation (NSF) EF-1253225 and DMS-1916395, and National Aeronautics and Space Administration's Carbon Monitoring System project. Banerjee was supported by NSF DMS-1513654, IIS-1562303, and DMS-1916349.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.envsoft.2019.104608.

References

Bakar, K.S., Kokic, P., Jin, H., 2015. Hierarchical spatially varying coefficient and temporal dynamic process models using spTDyn. J. Stat. Comput. Simul. 86 https://doi.org/10.1080/00949655.2015.1038267, 820-820.

Bakar, K.S., Kokic, P., Jin, H., 2015b. A spatio-dynamic model for assessing frost risk in south-eastern Australia. J. R. Stat. Soc. Ser. A C. URL 10.1111/rssc.12103.

Bakar, K.S., Kokic, P., Jin, H., 2017. Spatially Varying and Spatio-Temporal Dynamic Linear Models. R Package Version 2.0.

Bakar, K.S., Sahu, S.K., 2018. Spatio-Temporal Bayesian Modeling. R Package Version 3.3.

Bakka, H., Rue, H., Fuglstad, G.A., Riebler, A.I., Bolin, D., Illian, J., Krainski, E., Simpson, D.P., Lindgren, F.K., 2018. Spatial modelling with INLA: a review. ArXiv e-prints 10, e1443, 1802.06350. https://doi.org/10.1002/wics.1443.

Banerjee, S., Carlin, B.P., Gelfand, A.E., 2014. Hierarchical Modeling and Analysis for Spatial Data, second ed. Chapman & Hall/CRC, Boca Raton, FL.

Bivand, R., 2019. *CRAN Task View: Analysis Of Spatial Data*. 2019-02-25. https://cran.r-project.org/web/views/Spatial.html.

Bivand, R., Yu, D., 2017. Spgwr: Geographically Weighted Regression. R Package Version 0.6-32. https://CRAN.R-project.org/package=spgwr.

Brunekreef, B., Holgate, S.T., 2002. Air pollution and health. The Lancet 360 (9341), 1233–1242.

Bürgin, R., Ritschard, G., 2017. Coefficient-wise tree-based varying coefficient regression with vcrpart. Journal of Statistical Software, Articles 80 (6), 1–33. https://doi.org/10.18637/jss.v080.i06. ISSN 1548-7660.

Candiani, G., Carnevale, C., Finzi, G., Pisoni, E., Volta, M., 2013. A comparison of reanalysis techniques: applying optimal interpolation and ensemble kalman filtering to improve air quality monitoring at mesoscale. Sci. Total Environ. 458–460, 7–14, 0.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., Riddell, A., 2017. Stan: a probabilistic programming language. Journal of Statistical Software, Articles 76 (1), 1–32. https://doi.org/10.18637/jss.v076.i01. ISSN 1548-7660. https://www.jstatsoft.org/v076/i01.

Dagum, L., Menon, R., 1998. OpenMP: an industry standard API for shared-memory programming. Computational Science & Engineering, IEEE 5 (1), 46–55.

Datta, A., Banerjee, S., Finley, A., Hamm, N., Schaap, M., 2016. Nonseparable dynamic nearest neighbor Gaussian process models for large spatio-temporal data with an application to particulate matter analysis. Ann. Appl. Stat. 10 (3), 1286–1316. ISSN 1932-6157.

Denby, B., Schaap, M., Segers, A., Builtjes, P., Horalek, J., 2008. Comparison of two data assimilation methods for assessing PM10 exceedances on the European scale. Atmos. Environ. 42 (30), 7122–7134.

European Commission, 2015. European Union Air Quality Standards. http://ec.europa.eu/environment/air/quality/standards.htm.

Fan, J., Zhang, W., 2008. Statistical methods with varying coefficient models. Stat. Interface 1 (1), 179–195.

Finley, A., Banerjee, S., Gelfand, A., 2015. spBayes for large univariate and multivariate point-referenced spatio-temporal data models. Journal of Statistical Software, Articles 63 (13), 1–28. ISSN 1548-7660.

Fotheringham, A., Brunsdon, C., Charlton, M., 2002. Geographically Weighted Regression: the Analysis of Spatially Varying Relationships. Wiley, ISBN 9780471496168.

Gelfand, A.E., Ghosh, S.K., 1998. Model choice: a minimum posterior predictive loss approach. Biometrika 85 (1), 1–11.

Gelfand, A.E., Kim, H.J., Sirmans, C.F., Banerjee, S., 2003. Spatial modeling with spatially varying coefficient processes. J. Am. Stat. Assoc. 98 (462), 387–396.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., Rubin, D., 2013. Bayesian Data Analysis, Third Edition. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, ISBN 9781439840955.

Hamm, N., Finley, A., Schaap, M., Stein, A., 2015. A spatially varying coefficient model for mapping PM10 air quality at the European scale. Atmos. Environ. 102, 393–405.

Hastie, T., Tibshirani, R., 1993. Varying-coefficient models. J. R. Stat. Soc. Ser. B 55 (4), 757–796.

Hayfield, T., Racine, J.S., 2008. Nonparametric econometrics: the np package. J. Stat. Softw. 27 (5). http://www.jstatsoft.org/v27/i05/.

Heim, S., 2007. *Svcm: 2d and 3d Space-Varying Coefficient Models in R*. R Package Version 0.1.2.

Helske, J., 2019. Walker: Bayesian Regression with Time-Varying Coefficients. R Package Version 0.2.4-1. http://github.com/helske/walker.

Henderson, H.V., Searle, S.R., 1981. On deriving the inverse of a sum of matrices. SIAM Rev. 23 (1), 53–60.

Hoek, G., Krishnan, R.M., Beelen, R., Peters, A., Ostro, B., Brunekreef, B., Kaufman, J.D., 2013. Long-term air pollution exposure and cardio- respiratory mortality: a review. Environ. Health 12, 43.

Hothorn, T., Buehlmann, P., Kneib, T., Schmid, M., Hofner, B., 2018. *Mboost: Model-Based Boosting*. R Package Version 2.9-1. https://CRAN.R-project.org/package=mboost.

Lindgren, F., Rue, H., 2015. Bayesian spatial modelling with R-INLA. J. Stat. Softw. 63 (19), 1–25. http://www.jstatsoft.org/v63/i19/.
Loomis, D., Grosse, Y., Lauby-Secretan, B., El Ghissassi, F., Bouvard, V., Benbrahim-Tallaa, L., Guha, N., Baan, R., Mattock, H., Straif, S., 2013. The carcinogenicity of outdoor air pollution. Lancet Oncol. 14 (13), 1262–1263.
R Core Team, 2018. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.
Robert, C., Casella, G., 2004. Monte Carlo Statistical Methods. Springer Texts in Statistics, second ed. Springer-Verlag.
Roberts, G.O., Rosenthal, J.S., 2009. Examples of adaptive MCMC. J. Comput. Graph. Stat. 18 (2), 349–367.
Rue, H., Martino, S., Chopin, N., 2009. Approximate bayesian inference for latent Gaussian models using integrated nested laplace approximations (with discussion). J. R. Stat. Soc. B 71, 319–392.
Schaap, M., Timmermans, R.M.A., Roemer, M., Boersen, G.A.C., Builtjes, P., Sauter, F., Velders, G., Beck, J., 2008. The LOTOS-EUROS model: description, validation and latest developments. Int. J. Environ. Pollut. 32 (2), 270–290.
Spiegelhalter, D.J., Best, N.G., Carlin, B.P., van der Linde, A., 2001. Bayesian measures of model complexity and fit.
Stan Development Team, 2018. "RStan: the R Interface to Stan." R Package Version 2.18.2. http://mc-stan.org/.
van de Kassteele, J., Stein, A., 2006. A model for external drift kriging with uncertain covariates applied to air quality measurements and dispersion model output. Environmetrics 17 (4), 309–322.
Vihola, M., Helske, J., Franks, J., 2017. Importance sampling type estimators based on approximate marginal MCMC. ArXiv e-prints, 1609.02541.
Wood, S., 2017. Generalized Additive Models: an Introduction with R, 2 edition. Chapman and Hall/CRC.
Zhang, X., 2016. An Optimized BLAS Library Based on GotoBLAS2. https://github.com/xianyi/OpenBLAS/. Accessed 2015-06-01.