



## spTimer: Spatio-Temporal Bayesian Modeling Using R

**Khandoker Shuvo Bakar**  
Yale University

**Sujit K. Sahu**  
University of Southampton

---

### Abstract

Hierarchical Bayesian modeling of large point-referenced space-time data is increasingly becoming feasible in many environmental applications due to the recent advances in both statistical methodology and computation power. Implementation of these methods using the Markov chain Monte Carlo (MCMC) computational techniques, however, requires development of problem-specific and user-written computer code, possibly in a low-level language. This programming requirement is hindering the widespread use of the Bayesian model-based methods among practitioners and, hence there is an urgent need to develop high-level software that can analyze large data sets rich in both space and time.

This paper develops the package **spTimer** for hierarchical Bayesian modeling of stylized environmental space-time monitoring data as a contributed software package in the R language that is fast becoming a very popular statistical computing platform. The package is able to fit, spatially and temporally predict large amounts of space-time data using three recently developed Bayesian models. The user is given control over many options regarding covariance function selection, distance calculation, prior selection and tuning of the implemented MCMC algorithms, although suitable defaults are provided. The package has many other attractive features such as on the fly transformations and an ability to spatially predict temporally aggregated summaries on the original scale, which saves the problem of storage when using MCMC methods for large datasets. A simulation example, with more than a million observations, and a real life data example are used to validate the underlying code and to illustrate the software capabilities.

*Keywords:* Bayesian spatio-temporal modeling, Markov chain Monte Carlo, Gibbs sampling, autoregressive, predictive processes.

---

## 1. Introduction

Model-based Bayesian analysis methods are becoming popular for taking account of uncertainties in the analysis and spatial and temporal prediction of environmental space-time data.

Practitioners are increasingly benefiting from their ability to reduce uncertainty in the inference statements that arise from joint space-time modeling (Cressie and Wikle 2011). Bayesian methods are also popular because of their ability to combine information from several sources using melding or data fusion (Sahu, Gelfand, and Holland 2010). However, currently there is no suitable software package for Bayesian modeling and analysis of large space-time data. In this paper, our interest is on modeling and analyzing spatio-temporal *point-referenced* data (Banerjee, Carlin, and Gelfand 2004), where random observations are measured over time at a number of spatial locations, which vary continuously over a study region.

A number of R (R Core Team 2014) packages are available for modeling and analyzing spatial data, e.g., the packages **sp** (Pebesma and Bivand 2005) and **spacetime** (Pebesma 2012; Bivand, Pebesma, and Gómez-Rubio 2013). Furthermore, packages **gstat** (Pebesma 2004), **splm** (Millo and Piras 2012), **fields** (Nychka, Furrer, and Sain 2014) and **nlme** (Pinheiro, Bates, DebRoy, Sarkar, and R Core Team 2014) are able to fit spatial regression models and perform spatial interpolation-based on Kriging (Krige 1951). Some packages are also available for analyzing spatial point pattern data (Banerjee *et al.* 2004), see e.g., **spatial** (Ripley 2013; Venables and Ripley 2002) and **spatstat** (Baddeley and Turner 2005). However, none of these packages implement MCMC-based Bayesian modeling methods.

General purpose R software packages, such as **MCMCpack** (Martin, Quinn, and Park 2011), **MCMCglmm** (Hadfield 2010), and **BLR** (de los Campos and Perez Rodriguez 2014), are available for implementing Bayesian models. However, these are not suitable for analyzing spatially correlated data. Packages that can handle spatial correlations include **spBayes** (Finley, Banerjee, and Carlin 2007; Finley, Banerjee, and Gelfand 2015), **geoR** (Ribeiro and Diggle 2001), **geoRglm** (Christensen and Ribeiro 2002), **rjags** (Plummer 2014) and **R2WinBUGS** (Sturtz, Ligges, and Gelman 2005) which is an R interface to **WinBUGS** (Spiegelhalter, Thomas, Best, and Lunn 2003). However, these are not suitable and sometimes complicated for modeling data rich in both space and time, although the package **spBayes** can model spatially varying short-length time series data. These packages are not intended to handle large, e.g., more than a million, space-time data and these packages do not allow incorporation of popular models in the time series literature such as the auto-regressive models.

Non-Bayesian packages implementing the generalized additive models such as **gam** (Hastie 2014) and **mgcv** (Wood 2006) can also fit models for spatial data by implementing functional relationships between the response and the coordinates of the spatial locations, e.g., latitude and longitude. However, these modeling approaches are not process-based, i.e., do not incorporate random spatial and temporal processes, and we find that our process-based models implemented in **spTimer** have superior performance in out of sample spatial predictions, see Sections 4 and 5.

The main contribution of this paper is the development of the R package **spTimer** that is available from the Comprehensive R Archive Network at <http://CRAN.R-project.org/package=spTimer>. The package enables Bayesian modeling of regularly monitored point-referenced data obtained from a sparse spatial network of monitoring stations. Data at each monitoring station are obtained as a regular time series but may contain missing data. The package is also able to incorporate an arbitrary number of explanatory variables that may vary in both space and time. The residual spatio-temporal variation is modeled using three different recently developed modeling methods appropriate for analyzing space-time environmental monitoring data. All the inferences in this package and the paper are proposed to be under the Bayesian paradigm using MCMC methods.

Using Bayesian computation methods, the package **spTimer** is able to process, i.e., fit and predict, both spatially and temporally, large space-time data sets that may contain missing data. In so doing, the user is able to choose a particular covariance function from the Matérn family (Cressie 1993) for the underlying Gaussian process and also a suitable method for calculating distances between two locations. In addition, the user can select the hyper-parameters of the prior distributions and is also given the ability to choose the tuning parameters for the implemented MCMC algorithms. The package also allows the user to select one of the two possible (log and square-root) on the fly transformations for the response variable. Another attractive feature of the package is the ability to spatially predict temporally aggregated summaries, e.g., annual mean from daily data, on the original scale, which only requires storage of the annual aggregate, instead of the full length time series, for each prediction location at each MCMC iteration when we fit and predict for large data sets.

Users of **spTimer** only need to provide the model specifications in the high-level R language, but the main body of the code is developed using the C programming language, that is hidden from the user. This enables faster computation and better data handling capacity than what can be achieved by simply using R. However, once the MCMC iterations have been finished the output can be analyzed using any other contributed R package such as **coda** (Plummer, Best, Cowles, and Vines 2006). For model selection purposes, the package automatically reports the values of a predictive model selection criteria (Gelfand and Ghosh 1998). Many other utility functions for model validation and output analyzes are also provided. The main functions of the package **spTimer** are discussed in detail in Section 3.

The first of the three models (see Section 2) implemented in **spTimer** is a hierarchical nugget effect model together with an independent Gaussian process (GP) model at each time point. The Gaussian process implies a spatio-temporal random effect that captures the space-time interactions, see e.g., Cressie and Wikle (2011, Chapter 6). Overall, this model parallels the spatial random effect model in spatial only data analysis and naturally provides a very simple starting model in many investigations involving space-time data, see Banerjee *et al.* (2004); Diggle and Ribeiro (2007); and Gelfand, Diggle, Fuentes, and Guttorp (2010).

The second implemented model is the hierarchical auto-regressive model for space-time data developed by Sahu, Gelfand, and Holland (2007). An explicit auto-regressive term for the underlying true spatio-temporal process is assumed in a hierarchical set-up that includes the overall nugget effect. This model is included in the **spTimer** package because of its superior performance in modeling air-pollution data, see e.g., Cameletti, Ignaccolo, and Bande (2009) and Sahu and Bakar (2012a). This auto-regressive (AR) model is modified, as the third and final implemented model, to include the recently developed Gaussian predictive process approximation technique for handling large spatial and spatio-temporal data following Banerjee, Gelfand, Finley, and Sang (2008), Finley, Sang, Banerjee, and Gelfand (2009) and Sahu and Bakar (2012b). This last paper illustrates the capability of **spTimer** in handling and processing more than a million space-time observations within a reasonable amount of computing time in a standard personal computer.

This paper illustrates the package **spTimer** with two examples. The first is a simulation example (see Section 4) that is used for code verification and illustration of the software capabilities. The highlight of this example is that it simulates more than a million space-time observations from the third model-based on a Gaussian predictive process approximation. It then fits the model and illustrates prediction using MCMC methods. The code for the other two models are verified using a smaller simulation data set. A real data example, modeling

the daily 8-hour maximum ozone concentration in the months of July and August 2006 in the state of New York, is used for rapid illustration of the models and methods, see Section 5.

## 2. Spatio-temporal models

### 2.1. Preliminaries

The Bayesian spatio-temporal models can be represented in a hierarchical structure, where, according to Gelfand (2012), we specify distributions for data, process and parameters in three stages:

First	$[data process, parameter]$
Second	$[process parameter]$
Third	$[parameter]$

In the second stage, the process can add different levels, for example in Gaussian process (GP) models (Cressie and Wikle 2011; Gelfand *et al.* 2010) we have true underlying process in the first level and the spatio-temporal random effect in the second level of the hierarchy. Some illustrations are provided below in Section 2.2. Further examples, based on temporal processes, are given in Sections 2.3 and 2.4. In the third stage of the hierarchy we introduce the prior distribution of the parameters or hyper-parameters.

The models are described for time series data that are segmented using two different units of time, such as hours within days or days within years, to have extra flexibility in modeling and inputting data into the package. This enables us, for example, to model observed ozone concentration levels during the high ozone season (May to September in the United States) in each year for several years without having to model data for the remaining months in each year when ozone concentration levels are low and not harmful (Sahu and Bakar 2012b). However, by default, the package works for modeling data indexed by just one unit of time.

Let  $l$  and  $t$  denote the two units of time where  $l$  denotes the longer unit, e.g., year,  $l = 1, \dots, r$ , and  $t$  denotes the shorter unit, e.g., day,  $t = 1, \dots, T_l$  where  $r$  and  $T_l$  denote the total number of two time units, respectively. Let  $Z_l(\mathbf{s}_i, t)$  denote the observed point-referenced data and  $O_l(\mathbf{s}_i, t)$  be the true value corresponding to  $Z_l(\mathbf{s}_i, t)$  at site  $\mathbf{s}_i$ ,  $i = 1, \dots, n$  at time denoted by two indices  $l$  and  $t$ . Let  $\mathbf{Z}_{lt} = (Z_l(\mathbf{s}_1, t), \dots, Z_l(\mathbf{s}_n, t))^\top$  and  $\mathbf{O}_{lt} = (O_l(\mathbf{s}_1, t), \dots, O_l(\mathbf{s}_n, t))^\top$ . We shall denote all the observed data by  $\mathbf{z}$  and  $\mathbf{z}^*$  will denote all the missing data. Let  $N = n \sum_{l=1}^r T_l$  be the total number of observations to be modeled.

Throughout, the notation  $\boldsymbol{\epsilon}_{lt} = (\epsilon_l(\mathbf{s}_1, t), \dots, \epsilon_l(\mathbf{s}_n, t))^\top$  will be used to denote the so called nugget effect or the pure error term assumed to be independently normally distributed  $N(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I}_n)$ , where  $\sigma_\epsilon^2$  is the unknown pure error variance and  $\mathbf{I}_n$  is the identity matrix of order  $n$ . The spatio-temporal random effects will be denoted by  $\boldsymbol{\eta}_{lt} = (\eta_l(\mathbf{s}_1, t), \dots, \eta_l(\mathbf{s}_n, t))^\top$  and these will be assumed to follow  $N(\mathbf{0}, \Sigma_\eta)$  independently in time, where  $\Sigma_\eta = \sigma_\eta^2 S_\eta$ ,  $\sigma_\eta^2$  is the site invariant spatial variance and  $S_\eta$  is the spatial correlation matrix obtained from the, often used, general Matérn correlation function (Matérn 1986; Handcock and Stein 1993; Handcock and Wallis 1994) defined as:

$$\kappa(\mathbf{s}_i, \mathbf{s}_j; \phi, \nu) = \frac{1}{2^{\nu-1} \Gamma(\nu)} (2\sqrt{\nu} \|\mathbf{s}_i - \mathbf{s}_j\| \phi)^\nu K_\nu(2\sqrt{\nu} \|\mathbf{s}_i - \mathbf{s}_j\| \phi), \quad \phi > 0, \nu > 0, \quad (1)$$

where  $\Gamma(\nu)$  is the standard gamma function,  $K_\nu$  is the modified Bessel function of second kind with order  $\nu$ , and  $\|\mathbf{s}_i - \mathbf{s}_j\|$  is the distance between sites  $\mathbf{s}_i$  and  $\mathbf{s}_j$ . The parameter  $\phi$  controls the rate of decay of the correlation as the distance  $\|\mathbf{s}_i - \mathbf{s}_j\|$  increases and the parameter  $\nu$  controls smoothness of the random field (Banerjee *et al.* 2004; Cressie 1993). The package **spTimer** allows several possibilities regarding estimation of the correlation parameters  $\phi$  and  $\nu$  that range from fixing them (point mass prior) to estimating them by assuming suitable prior distributions, see Section 2.5 for further details. In addition, **spTimer** is also able to incorporate the spherical correlation function, see Banerjee *et al.* (2004).

In the development below, we assume that there are  $p$  covariates, including the intercept, denoted by the  $n \times p$  matrix  $\mathbf{X}_{lt}$ . Some of these covariates may vary in space and time. The notation  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)$  will be used to denote the  $p \times 1$  vector of regression coefficients. We shall use the generic notation  $\boldsymbol{\theta}$  to denote all the parameters.

## 2.2. GP model specification

The independent Gaussian process (GP) model is specified hierarchically by:

$$\mathbf{Z}_{lt} = \mathbf{O}_{lt} + \boldsymbol{\epsilon}_{lt}, \quad (2)$$

$$\mathbf{O}_{lt} = \mathbf{X}_{lt}\boldsymbol{\beta} + \boldsymbol{\eta}_{lt}, \quad (3)$$

for each  $l = 1, \dots, r$  and  $t = 1, \dots, T_l$ , where we assume that  $\boldsymbol{\epsilon}_{lt}$  and  $\boldsymbol{\eta}_{lt}$  are independent and each are normally distributed with their respective parameters as given in Section 2.1. Let  $\mathbf{O}$  denote all the random effects,  $\mathbf{O}_{lt}$ ,  $l = 1, \dots, r$  and  $t = 1, \dots, T_l$ . Let  $\boldsymbol{\theta} = (\boldsymbol{\beta}, \sigma_\epsilon^2, \sigma_\eta^2, \phi, \nu)$  denote all the parameters of this model and let  $\pi(\boldsymbol{\theta})$  denote the prior distribution that we shall specify later. The logarithm of the joint posterior distribution of the parameters and the missing data for this GP model is given by:

$$\begin{aligned} \log \pi(\boldsymbol{\theta}, \mathbf{O}, \mathbf{z}^* | \mathbf{z}) &\propto -\frac{N}{2} \log \sigma_\epsilon^2 - \frac{1}{2\sigma_\epsilon^2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{Z}_{lt} - \mathbf{O}_{lt})^\top (\mathbf{Z}_{lt} - \mathbf{O}_{lt}) - \frac{\sum_{l=1}^r T_l}{2} \log |\sigma_\eta^2 S_\eta| \\ &\quad - \frac{1}{2\sigma_\eta^2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta})^\top S_\eta^{-1} (\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta}) + \log \pi(\boldsymbol{\theta}). \end{aligned} \quad (4)$$

The prior distribution  $\pi(\boldsymbol{\theta})$  is specified in Section 2.5 and the full conditional distributions of the parameters, required for Gibbs sampling, are provided in Appendix A.

## 2.3. AR model specification

Following Sahu *et al.* (2007), we specify the hierarchical AR models as follows:

$$\mathbf{Z}_{lt} = \mathbf{O}_{lt} + \boldsymbol{\epsilon}_{lt}, \quad (5)$$

$$\mathbf{O}_{lt} = \rho \mathbf{O}_{lt-1} + \mathbf{X}_{lt}\boldsymbol{\beta} + \boldsymbol{\eta}_{lt}, \quad (6)$$

for all  $l$  and  $t$ ; where,  $\rho$  denotes the unknown temporal correlation parameter assumed to be in the interval  $(-1, 1)$ . Obviously, for  $\rho = 0$ , these models reduce to the GP models described above in Section 2.2. We continue to assume the Gaussian distributions, introduced in Section 2.1, for  $\boldsymbol{\epsilon}_{lt}$  and  $\boldsymbol{\eta}_{lt}$  for all values of  $l$  and  $t$ .

The auto-regressive models require specification of the initial term  $\mathbf{O}_{l0}$  for each  $l = 1, \dots, r$ . Here we specify an independent spatial model for each  $\mathbf{O}_{l0}$  with mean  $\boldsymbol{\mu}_l$  and the covariance

matrix  $\sigma_l^2 S_0$  where the correlation matrix  $S_0$  is obtained using the Matérn correlation function in Equation (1) with the same set of correlation parameters  $\phi$  and  $\nu$  for  $\boldsymbol{\eta}_{lt}$ .

Let  $\boldsymbol{\theta}$  denote all the parameters, i.e.,  $\boldsymbol{\theta} = (\boldsymbol{\beta}, \rho, \sigma_\epsilon^2, \sigma_\eta^2, \phi, \nu, \boldsymbol{\mu}_l, \sigma_l^2)$  and we also suppose that  $\mathbf{O}$  contains all the parameters  $\mathbf{O}_{lt}$  for  $l = 1, \dots, r$ ,  $t = 0, \dots, T_l$ . The logarithm of the joint posterior distribution of the parameters and the missing data is now given by:

$$\begin{aligned} \log \pi(\boldsymbol{\theta}, \mathbf{O}, \mathbf{z}^* | \mathbf{z}) &\propto -\frac{N}{2} \log \sigma_\epsilon^2 - \frac{1}{2\sigma_\epsilon^2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{z}_{lt} - \mathbf{O}_{lt})^\top (\mathbf{z}_{lt} - \mathbf{O}_{lt}) - \frac{\sum_{l=1}^r T_l}{2} \log |\sigma_\eta^2 S_\eta| \\ &\quad - \frac{1}{2\sigma_\eta^2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{O}_{lt} - \rho \mathbf{O}_{lt-1} - \mathbf{X}_{lt} \boldsymbol{\beta})^\top S_\eta^{-1} (\mathbf{O}_{lt} - \rho \mathbf{O}_{lt-1} - \mathbf{X}_{lt} \boldsymbol{\beta}) \\ &\quad - \frac{1}{2} \sum_{l=1}^r \log |\sigma_l^2 S_0| - \frac{1}{2} \sum_{l=1}^r \frac{1}{\sigma_l^2} (\mathbf{O}_{l0} - \boldsymbol{\mu}_l)^\top S_0^{-1} (\mathbf{O}_{l0} - \boldsymbol{\mu}_l) + \log \pi(\boldsymbol{\theta}). \end{aligned} \quad (7)$$

As for the GP models of previous section, the prior distributions are specified in Section 2.5. Full conditional distributions are provided in Appendix B.

## 2.4. Spatio-temporal models based on GPP

These models are based on the recent work by Sahu and Bakar (2012b) and the main idea here is to define the random effects  $\eta_l(\mathbf{s}_i, t)$  at a smaller number,  $m$ , of locations, called the knots, and then use kriging to predict those random effects at the data and prediction locations. Here, an AR model is only assumed for the random effects at the knot locations and not for all the random effects at the observation locations. At the top level we assume the model:

$$\mathbf{Z}_{lt} = \mathbf{X}_{lt} \boldsymbol{\beta} + A \mathbf{w}_{lt} + \boldsymbol{\epsilon}_{lt}, \quad (8)$$

for all  $l$  and  $t$ , where  $A = C S_\eta^{-1}$  and  $C$  denotes the  $n$  by  $m$  cross-correlation matrix between the random effects at the  $n$  observation locations and  $m$  knot locations,  $\mathbf{s}_1^*, \dots, \mathbf{s}_m^*$ , and  $S_\eta$  is the  $m$  by  $m$  correlation matrix of the  $m$  random effects  $\mathbf{w}_{lt}$ . We specify  $\mathbf{w}_{lt}$  at the knots conditionally given  $\mathbf{w}_{lt-1}$  as:

$$\mathbf{w}_{lt} = \rho \mathbf{w}_{lt-1} + \boldsymbol{\eta}_{lt}, \quad (9)$$

for all  $l$  and  $t$ , where  $\boldsymbol{\eta}_{lt} \sim N(\mathbf{0}, \Sigma_\eta)$  independently,  $\Sigma_\eta = \sigma_\eta^2 S_\eta$ . Note that, here  $\Sigma_\eta$  is an  $m \times m$  matrix which is of much lower dimensionality than the same for two previous models GP and AR since we assume that  $m \ll n$ .

The auto-regressive models are completed by the assumption for the initial conditions,  $\mathbf{w}_{l0} \sim N(\mathbf{0}, \sigma_l^2 S_0)$  independently for each  $l = 1, \dots, r$ , where the correlation matrix  $S_0$  is obtained by using the Matérn correlation function in Equation (1) with decay parameter  $\phi$ . Let  $\mathbf{w}$  denote the random effects  $\mathbf{w}_{lt}$  for  $l = 1, \dots, r$  and  $t = 0, 1, \dots, T_l$ . Let  $\boldsymbol{\theta}$  denote all the parameters  $\boldsymbol{\beta}, \rho, \sigma_\epsilon^2, \sigma_w^2, \phi, \nu, \sigma_l^2, l = 1, \dots, r$ . The logarithm of the joint posterior distribution of the parameters and the missing data is given by:

$$\begin{aligned} \log \pi(\boldsymbol{\theta}, \mathbf{w}, \mathbf{z}^* | \mathbf{z}) &\propto -\frac{N}{2} \log \sigma_\epsilon^2 - \frac{1}{2\sigma_\epsilon^2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{z}_{lt} - \mathbf{X}_{lt} \boldsymbol{\beta} - A \mathbf{w}_{lt})^\top (\mathbf{z}_{lt} - \mathbf{X}_{lt} \boldsymbol{\beta} - A \mathbf{w}_{lt}) \\ &\quad - \frac{m \sum_{l=1}^r T_l}{2} \log \sigma_\eta^2 - \frac{\sum_{l=1}^r T_l}{2} \log |S_\eta| - \frac{1}{2\sigma_\eta^2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{w}_{lt} - \rho \mathbf{w}_{lt-1})^\top S_\eta^{-1} \end{aligned}$$



$$\begin{aligned}
& (\mathbf{w}_{lt} - \rho \mathbf{w}_{lt-1}) - \frac{m}{2} \sum_{l=1}^r \log \sigma_l^2 - \frac{r}{2} \log |S_0| - \frac{1}{2} \sum_{l=1}^r \frac{1}{\sigma_l^2} \mathbf{w}_{l0} S_0^{-1} \mathbf{w}_{l0} \\
& + \log \pi(\boldsymbol{\theta}).
\end{aligned} \tag{10}$$

As previously in Sections 2.2 and 2.3, we specify the prior distributions in Section 2.5. The full conditional distributions for Gibbs sampling are provided in Appendix C.

## 2.5. Prior distributions

The Bayesian model for each of the above three specifications is completed by assuming suitable prior distributions for the underlying parameters. For simplicity and convenience, we group the model parameters of different spatial processes and data into three different types depending on whether those describe: the mean, the variance or the correlation. All the parameters describing the mean, e.g.,  $\beta, \rho$ , other than the random effects, are given independent normal prior distributions, where the user can specify the means ( $\mu_\beta, \mu_\rho$ ) and variances ( $\delta_\beta^2, \delta_\rho^2$ ). In our illustrations in Section 3, we have taken all those means to be 0 and variances to be  $10^{10}$ , that correspond to our assumption of flat prior distributions. We also define an independent normal prior distribution with mean 0 and  $\sigma_\mu^2$ , assumed to be  $10^{10}$  in our illustrations, for each component of the  $n$ -dimensional vector  $\boldsymbol{\mu}_l$  in the autoregressive model.

The prior distribution for a typical precision (inverse of variance) parameter is specified through a gamma distribution with mean  $a/b$  and variance  $a/b^2$ . The user can specify any suitable values for  $a$  and  $b$ , but in our illustrations we have chosen  $a = 2$  and  $b = 1$  to have a proper prior distribution for any variance component that will guarantee a proper posterior distribution (Gelman, Carlin, Stern, and Rubin 2004).

There is a large literature regarding the identifiability and the consistency of the parameters describing the correlation in a Gaussian process; see for example, Stein (1999) and Zhang (2004). These problems manifest themselves in the Bayesian estimation literature as well, see e.g., Sahu, Gelfand, and Holland (2006), who use an empirical Bayes (EB) approach. To facilitate estimation using this EB approach, **spTimer** allows the user to run Gibbs sampling by fixing the correlation parameters  $\phi$  and  $\nu$  so that a grid search can be performed to find the optimal values of these parameters.

Full Bayesian estimation of  $\phi$  and  $\nu$  is also allowed in **spTimer** corresponding to discrete uniform prior distributions for both these parameters. In all our illustrations the smoothing parameter  $\nu$  is estimated using a discrete uniform prior distribution taking values from 0 to 1.5 with the increment 0.05. In addition to a discrete uniform distribution, we also allow a Gamma prior distribution with suitable hyper-parameter values for the decay parameter  $\phi$ .

In practice, in any Bayesian analysis, the sensitivity with respect to the chosen distribution must be studied, and the **spTimer** package that we have developed here makes it easy to do so without much programming effort.

## 2.6. Model fitting

All the models are fitted using Gibbs sampling (Gelfand and Smith 1990). The conjugate prior distributions assumed for all the parameters except for the  $\phi$  and  $\nu$  enable standard conjugate sampling from the full conditional distributions. These details are given in Appendices A, B,

and C. Missing data values are sampled from their conditional distributions at each iteration of the Gibbs sampler.

The full conditional distributions of the correlation parameters  $\phi$  and  $\nu$  are non-standard. The package provides two options for sampling these parameters corresponding to the two different prior distributions. The full conditional distribution will be discrete, and hence easy to sample from, if a discrete uniform prior distribution has been assumed for them. The second option, only allowed for the decay parameter  $\phi$ , is to assign a continuous uniform prior distribution over an interval or a Gamma prior distribution and then to use the random-walk Metropolis-Hastings algorithm to draw samples from it. The proposal distribution is taken as the normal distribution with the mean at the current value and the variance  $\sigma_p^2$  which is tuned to have an acceptance rate between 15 to 40%, see Gelman *et al.* (2004) for theoretical justifications. The Metropolis algorithm is implemented on the log-scale for  $\phi$ , i.e., we work with the density of  $\log(\phi)$  instead of  $\phi$  since the support of the normal proposal distribution is the real line. In this sampling scheme, keeping  $\phi$  within a range, as may be required by the assumed prior distribution, is trivial since any proposal value outside the range is rejected forthwith.

The quality of the model fit and its predictive abilities are assessed by calculating the predictive model choice criteria (PMCC, Gelfand and Ghosh 1998) which is given by,

$$\text{PMCC} = \sum_{i=1}^n \sum_{l=1}^r \sum_{t=1}^{T_l} \left\{ \text{E}(Z_l(\mathbf{s}_i, t)_{\text{rep}} - z_l(\mathbf{s}_i, t))^2 + \text{Var}(Z_l(\mathbf{s}_i, t)_{\text{rep}}) \right\}, \quad (11)$$

where,  $Z_l(\mathbf{s}_i, t)_{\text{rep}}$  denotes a future replicate of the data  $z_l(\mathbf{s}_i, t)$ . The first term in the PMCC assesses the goodness of fit and second term is a penalty term for model complexity. The PMCC, justified using a squared error loss function, is most suitable for comparing Bayesian hierarchical models that involve a first stage Gaussian model. The values of the criteria are estimated by sampling from the posterior predictive distributions discussed below.

## 2.7. Prediction details

Spatial prediction at a new location and temporal prediction at future time points proceed with the posterior predictive distribution for  $Z_l(\mathbf{s}_0, t')$ , where  $\mathbf{s}_0$  denotes a new location and  $t'$  is a time point, which can be in the future. The posterior predictive distribution for  $Z_l(\mathbf{s}_0, t')$  is obtained by integrating over the parameters with respect to the joint posterior distribution as:

$$\begin{aligned} \pi(Z_l(\mathbf{s}_0, t')|\mathbf{z}) &= \int \pi(Z_l(\mathbf{s}_0, t')|O_l(\mathbf{s}_0, t'), \sigma_\epsilon^2) \times \pi(O_l(\mathbf{s}_0, t')|\boldsymbol{\theta}, \mathbf{O}, \mathbf{z}^*) \\ &\quad \times \pi(\boldsymbol{\theta}, \mathbf{O}, \mathbf{z}^*|\mathbf{z}) dO_l(\mathbf{s}_0, t') d\mathbf{O} d\boldsymbol{\theta} d\mathbf{z}^*. \end{aligned} \quad (12)$$

Note that for the GPP models in Section 2.4 we replace the random effects  $O_l(\mathbf{s}, t')$  by  $w_l(\mathbf{s}, t')$ . Predictions are obtained by composition sampling. First, a random sample  $\boldsymbol{\theta}^{(j)}, \mathbf{O}^{(j)}$  is drawn from the posterior distribution  $\pi(\boldsymbol{\theta}, \mathbf{O}, \mathbf{z}^*|\mathbf{z})$  using the details in the model fitting described in Sections 2.2, 2.3, 2.4. Then, Bayesian kriging is applied to draw a sample,  $O_l^{(j)}(\mathbf{s}_0, t')$  from the conditional distribution of  $O_l(\mathbf{s}_0, t')$  given  $O_l(\mathbf{s}_1, t), \dots, O_l(\mathbf{s}_n, t)$ . Finally a sample  $Z_l^{(j)}(\mathbf{s}_0, t')$  is drawn from the top level model  $\pi(Z_l(\mathbf{s}_0, t')|O_l^{(j)}(\mathbf{s}_0, t'), \sigma_\epsilon^{2(j)})$ .

At the end of the MCMC run, the samples  $Z_l^{(j)}(\mathbf{s}_0, t')$ ,  $j = 1, \dots, J$  where  $J$  is a large number, are summarized to make predictive inference. If a transformation, such as the log or the square



Function	Description
<code>spT.Gibbs</code>	For model fitting using Gibbs sampling approach. It can also predict simultaneously if suitable options are turned on.
<code>predict.spT</code> or <code>predict</code>	Using output from <code>spT.Gibbs</code> this is able to perform prediction, either spatially or temporally or both.

Table 1: Two main functions in `spTimer`.

root, has been applied then the posterior predictive samples  $Z_l^{(j)}(\mathbf{s}_0, t')$  are first transformed back to the original scale before their summaries are calculated. Further, details regarding these predictions are provided in Bakar (2012) for the GP models, Sahu *et al.* (2007) for the AR models and Sahu and Bakar (2012b) for the GPP-based models.

### 3. Main functions in `spTimer`

There are two main functions in the `spTimer` package, namely, `spT.Gibbs` for model fitting and `predict` to obtain spatial and temporal predictions based on the fitted models. Table 1 provides a snapshot of these two main functions. In the following sub-sections we discuss these functions more elaborately.

#### 3.1. `spT.Gibbs`

The function `spT.Gibbs` is used to fit all three models using the Gibbs sampling approach and it takes a number of arguments that define and control its behavior. For example, the argument `formula` is used to specify the linear part of the model. It is an object of class `formula`, which is same as that for the `lm` function to fit linear regression models in R.

The argument `data` provides the data frame to be used for model fitting. The package also has the capability to support data classes `STFDF` of `spacetime` and `SpatialPointsDataFrame` of `sp`. The data set must be first ordered by the location index, and for each location the data must be ordered by time. Time series data with more than one segment, for example,  $T_l$  unit (e.g., daily) observations in the  $l$ th segment ( $l = 1, \dots, r$ ) (e.g., month), must be ordered first by the longer unit (e.g., month) and then by the shorter unit (e.g., day). The total length of the time series at each location,  $\sum_{l=1}^r T_l$ , must be the same for each location and the data rows in each site must correspond to the same basic time unit.

Varying segment length time series are also allowed, e.g.,  $T_1 = 31$ ,  $T_2 = 28$ ,  $T_3 = 31$ , and  $T_4 = 30$  when it is intended to model daily data for the first four calendar months ( $r = 4$ ) in a year which is not a leap year. Moreover, all sites must have data rows for the exact same total number of time units and the rows for the shorter time units in each site must correspond to those for all other sites. This is to prohibit passing of irregularly observed time series to the `data` argument, i.e., in the current example one observation site cannot replace, for example, 2 days of January data by data for two extra days in May when all the other sites have daily data for the first four months. However, it is possible to have missing observations for the response variable and the missing values must be denoted by the standard `NA` identifier. No missing data values are allowed for the covariates since handling of such situations requires additional modelling. Details on how to define the time segments are provided in Section 3.3.

A typical code for the GP model is written as:

```

spT.Gibbs(formula, data = parent.frame(), model = "GP", time.data = NULL,
  coords, knots.coords, newcoords = NULL, newdata = NULL, priors = NULL,
  initials = NULL, nItr = 5000, nBurn = 1000, report = 1, tol.dist = 0.05,
  distance.method = "geodetic:km", cov.fnc = "exponential",
  scale.transform = "NONE", annual.aggrn = "NONE",
  spatial.decay = spT.decay(distribution = "FIXED"))

```

The `model` argument specifies the model to be fitted which can be any one of the three: GP, AR, and GPP with GP being the default. In `spT.Gibbs` the argument `priors` specify the hyperparameter values of the prior distributions. The default value of this argument is `NULL` when automatic values are chosen. More details regarding this are provided in Section 3.3.

The required argument `coords` can be supplied in different formats. It can be specified as an  $n \times 2$  matrix or data frame containing the co-ordinates of the  $n$  spatial locations, or as a formula object defining the coordinates, e.g., `coords = ~ Longitude + Latitude`. The optional argument `knots.coords` is used only for the GPP model and, must be provided as an  $m \times 2$  matrix of coordinates of the  $m$  ( $< n$ ) knot locations.

The two optional arguments `newcoords` and `newdata` need to be provided if it is intended to do model fitting and prediction simultaneously. The argument `newcoords` provides the new coordinate points where we want to predict and `newdata` must contain the values of the covariates at the prediction locations. No predictions are performed if these two arguments are omitted. In that case, predictions can still be performed after model fitting by calling the `predict` function, see Section 3.2.

The argument `initials` specifies the starting values for the model parameters. Default initial values can be chosen by specifying the option `initials = NULL`, and in that case the following values will be set:  $\sigma_\eta^2 = 0.1$ ,  $\sigma_\epsilon^2 = 0.01$ . The default initial value for the spatial decay parameter,  $\phi$ , is set as  $-\log(0.05)/d_{\max} \approx 3/d_{\max}$  (Finley *et al.* 2007), where  $d_{\max}$  is the maximum distance calculated from the coordinates of the model fitting locations, which ensures that the effective spatial range (the distance by which the spatial correlation becomes negligible) is  $d_{\max}$ . The initial values for the regression parameters and the auto-regressive parameter are obtained by fitting a simple linear model using the `lm` function.

The arguments `nItr`, `nBurn`, and `report` control the running of the MCMC algorithm where: `nItr` specifies the total number of iterations, `nBurn` denotes the number of burn-in iterations to discard, and `report` allows how many reports of progress of MCMC to print on screen.

The `distance.method` argument allows the distance between any two locations to be calculated using "geodetic:km" (the default), "geodetic:mile" and "euclidean" for distance in kilometers, miles and Euclidean unit respectively. Related to this is the `tol.dist`, which defaults to 0.05, that allows the user to avoid modelling locations which are less than the `tol.dist` away. This is to avoid singularity in the covariance matrices. The argument `cov.fnc` provides the choice of the spatial covariance function and can take any one of the values: "exponential" (the default), "gaussian", "spherical", and "matern". The handling of the spatial decay parameters is discussed in Section 3.3.

The argument `scale.transform` specifies the on-the-fly transformation for the response variable and it can take one of the three possible values "SQRT", "LOG" and "NONE" with the last one being the default. Note that all the predictions will be performed on the original scale of the data and it is not possible to use any on-the-fly transformation for the explanatory variables.

An optional argument `annual.aggrn` is also used only if the predictions are to be made using `spT.Gibbs`. This argument specifies the required type of time series aggregation in the predictions and, currently it can take any of the values: `"ave"` for annual average, `"an4th"` for obtaining the annual 4th highest value and `"NONE"` for no annual aggregates. For example, if dataset has 365 daily observations in each of 10 years, then the use of `annual.aggrn = "ave"` yields the 10 annual averages at each MCMC iteration without having to store 3650 iterates of the daily values. Thus, this argument helps to solve the storage problem when it is of interest to predict the aggregated summaries rather than the individual atomic space-time data.

### 3.2. predict

The function `predict` is used to get spatial and temporal predictions based on the results obtained from the routine `spT.Gibbs`. The two required arguments of `predict` are the `newcoords`, which must contain the new coordinate points where we want to predict and `newdata`, which contains the values of the covariates. The argument `type` in `predict` specifies the type of prediction the user wants to make which can be either `"spatial"` or `"temporal"`. If the value is `"spatial"` then only spatial prediction will be performed at the `newcoords` which must be different from the fitted sites provided by the `coords` argument. When the `"temporal"` option is specified then forecasting will be performed and in this case the `newcoords` may also contain elements of the fitted sites in which case only temporal forecasting beyond the last fitted time point will be performed.

### 3.3. Some other useful functions

The package `spTimer` includes several utility functions, e.g., `spT.time`, `spT.priors`, `spT.decay`, `spT.initials`, and `spT.validation` for performing various important tasks.

The function `spT.time` is used to specify the temporal structure of the data that can be used in `spT.Gibbs`. It has two arguments: (i) `t.series` that specifies the number of observations in each segment of the time series and (ii) `segments` that specifies the total number of segments. For example, to model data for 5 years with 30 days in each year we call the function `spT.time(t.series = 30, segments = 5)` and send the output to `spT.Gibbs`. The default value for `segments` is 1 and this is what should be used for simple time series modelling problems where time is described by one unit only, e.g., day.

The package can also handle unequal length time segments which can be specified by a vector valued `t.series` argument in `spT.time`. For example, to model daily data for the first four calendar months in a non-leap year, the appropriate code is `spT.time(t.series = c(31, 28, 31, 30), segments = 4)`. However, in this paper we only illustrate with equal segment length time series data.

The `spT.priors` function is used to define the hyper-parameter values of the prior distributions. As the parameters are model dependent, the call to this function requires a `model` argument and then a list of prior distributions for the associated parameters. The abbreviation `"Gamm"` denotes the gamma prior distribution, while the abbreviation `"Norm"` denotes the normal prior distribution. For example, the call

```
R> prior <- spT.priors(model, inv.var.prior = Gamm(a = 2, b = 1),
+   beta.prior = Norm(0, 10^10), rho.prior = Norm(0, 10^10))
```

will specify independent gamma prior distribution with parameters 2 and 1 for the inverse of the each variance component, and each of the regression parameters  $\beta$  and the autoregressive parameter  $\rho$  will be assigned an independent normal prior distribution with mean 0 and variance  $10^{10}$ . A default proper prior with large variance will be assumed if prior for any parameter is missing.

The `spT.decay` function is used to assign prior distribution for the spatial decay parameter  $\phi$  in one of the following three possible ways:

1. *Fixed:* This choice fixes  $\phi$  at a particular value, which is achieved by writing `distribution = "FIXED"` in the argument. For example, for fixing  $\phi$  at 0.01 we write:

```
spatial.decay = spT.decay(distribution = "FIXED", value = 0.01)
```

The default initial value,  $3/d_{\max}$ , will be assumed if the particular value is not provided. This "FIXED" option is the package default.

2. *Uniform distribution:* This option corresponds to assuming a discrete uniform prior for  $\phi$  in a specified interval. A typical specification is provided below:

```
spatial.decay = spT.decay(distribution=Unif(.01,.02), npoints=5)
```

where the `npoinst` argument specifies the number of support points in the prior distribution under this option. Default value for `npoinst` is 5. At the moment it is not possible to specify a continuous uniform distribution as the prior for the decay parameter.

3. *Gamma distribution:* The gamma prior distribution for  $\phi$  can be assumed by the statement:

```
spatial.decay = spT.decay(distribution = Gamm(a = 2, b = 1),
  tuning = 0.08)
```

where the `tuning` parameter specifies the standard deviation of the normal proposal distribution centered at the current value for the random-walk Metropolis sampling algorithm implemented for sampling  $\phi$  on the log-scale.

The function `spT.initials` is used to gather initial values for the parameters in the model.

The function `spT.validation` calculates the following validation criteria: mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), relative bias (rBIAS), and relative mean separation (rMSEP).

$$\begin{aligned} \text{MSE} &= \frac{1}{m} \sum_{i=1}^m (\hat{z}_i - z_i)^2, & \text{MAE} &= \frac{1}{m} \sum_{i=1}^m |\hat{z}_i - z_i|, \\ \text{MAPE} &= \frac{1}{m} \sum_{i=1}^m |(\hat{z}_i - z_i)/z_i|, & \text{BIAS} &= (\hat{z}_i - z_i), \\ \text{rBIAS} &= \frac{1}{m\bar{z}} \sum_{i=1}^m (\hat{z}_i - z_i), & \text{rMSEP} &= \sum_{i=1}^m (\hat{z}_i - z_i)^2 / \sum_{i=1}^m (\bar{z}_p - z_i)^2, \end{aligned}$$

where,  $m$  is the total number of observations we want to validate,  $z_i$  is the data indexed by  $i$ ,  $\hat{z}_i$  is the prediction value,  $\bar{z}$  and  $\bar{z}_p$  are the arithmetic mean of the observations and predictions respectively.

## 4. Simulation study

The main purpose of this example is to validate the body of code underpinning the package **spTimer**. The code for carrying out the main inference tasks: estimation of model parameters and predictions are validated for each of three models GP, AR and GPP. This section reports the results from experiments on larger data sets using GPP-based models. We examine different scenarios, for example an intercept only model and a model with four covariates to test inference of the regression ( $\beta$ ) parameters. We also provide a sensitivity analysis for different signal-to-noise ratios and compare the performance of **spTimer** with **spBayes** and **mgcv**, which adopts a non-Bayesian framework. The code lines for doing this analysis are provided in the accompanying R file, `v63i15.R`.

### 4.1. Simulation design and results

The spatial domain of the simulation study is taken as a square ranging from zero to 1000 units, and for fitting GP and AR models we suppose that the sampling locations form a regular grid of  $12 \times 12 = 144$  points (see Figure 1(a)) inside the square. For the GPP-based approximation model we consider  $55 \times 55 = 3025$  grid points inside the square, that is moderately large (see Figure 1(b)). The number of knots are defined for the GPP-based models is a  $10 \times 10$  square grid inside the range. The temporal domain is taken as 365 days in a year. Hence, for GP and AR models we obtain 52,560 ( $= 144 \times 365$ ) observations in total, and for the GPP model we generate a data set with 1,104,125 ( $= 3025 \times 365$ ) observations.

Out of these data locations (144 for the GP and AR models; and 3025 for the GPP-based model), we randomly choose 10% locations, as validation sites and model data from the remaining 90% sites. Hence, we model data from 129 locations for the GP and AR models

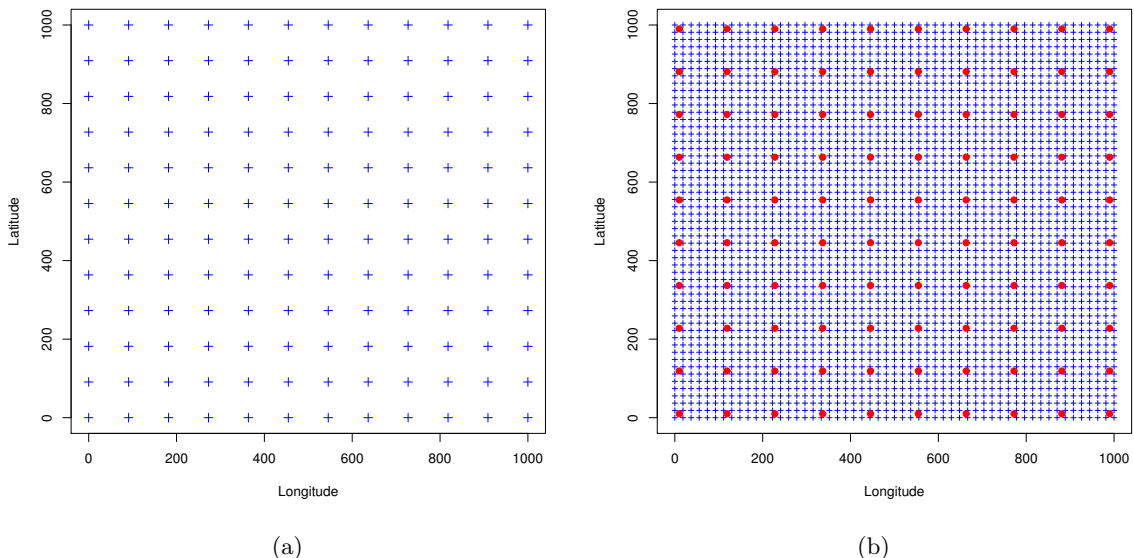


Figure 1: (a) A representation of the 144 grid locations (+) used for simulating the data for GP and AR models. (b) 3025 grid locations (+) are used for data simulation for the GPP-based approximation model and 100 knot points are superimposed using (red) solid circles.

Model	Parameter	True Value	Median	95% Interval
GP	$\beta_0$	5.0	4.99	(4.96, 5.03)
	$\beta_1$	2.0	1.95	(1.83, 2.11)
	$\beta_2$	1.0	0.94	(0.80, 1.27)
	$\beta_3$	0.5	0.52	(0.46, 0.57)
AR	$\beta_0$	5.0	5.10	(4.98, 5.27)
	$\beta_1$	2.0	1.98	(1.81, 2.30)
	$\beta_2$	1.0	1.05	(0.78, 1.21)
	$\beta_3$	0.5	0.47	(0.41, 0.55)
	$\rho$	0.2	0.19	(0.15, 0.27)
GPP	$\beta_0$	5.0	4.89	(4.75, 5.13)
	$\beta_1$	2.0	1.90	(1.77, 2.07)
	$\beta_2$	1.0	0.96	(0.81, 1.13)
	$\beta_3$	0.5	0.46	(0.38, 0.55)
	$\rho$	0.2	0.17	(0.09, 0.26)

Table 2: True values of the parameters and their estimates using the summary statistics of the MCMC samples for all three models. The column Median represents the posterior median and a 95% credible interval is also provided for each of the parameters.

and validate them for the set aside data from the remaining 15 locations. Similarly for the GPP-based models we set aside data from 300 locations for validations and model data from the remaining 2725 locations. We also assume that 5% data are missing at random for the GP, AR and GPP-based models. This is done to test the missing data handling capabilities of the Bayesian approach which is very common in practice. The Euclidean distance is used in the simulation study while the real life example in the next section uses the geodetic distance for a geographic spatial domain. We illustrate throughout using the exponential covariance function ( $\nu = 0.5$ ) for both simulation and model fitting, although we have validated using all other covariance functions.

Three different data sets are simulated from the three models using the following values of the model parameters.

We assume that four covariate effects are captured including the intercept term, and take the true value of the parameters as  $\beta_0 = 5.0$ ,  $\beta_1 = 2.0$ ,  $\beta_2 = 1.0$  and  $\beta_3 = 0.5$ . The covariates  $x_1, \dots, x_3$  are generated from the standard normal distribution. The spatial effect variance, i.e., the signal  $\sigma_\eta^2$  under GP, AR and GPP-based models is assumed to be randomly chosen from the uniform distribution in  $(0, 1)$  and the pure error (or nugget effect) variance is set at  $\sigma_\epsilon^2 = 0.01$ . We have also provided a sensitivity analysis of the signal-to-noise ratio later in this section. The spatial decay parameter  $\phi$  is assumed to be 0.003 that implies high spatial correlation even at large distances. The temporal auto-correlation parameter,  $\rho$  is set at 0.2 for the GP and GPP-based models. This moderate value was deemed to be enough for simulating space-time dependent data over and above the previously assumed high spatial correlation. For the AR model, the initial mean and variance for  $O_l(\mathbf{s}_i, 0)$  are taken to be 5.0 and 0.5 for  $l = 1$ . The number of knots for the GPP-based model is taken to be 100 (see Figure 1(b) for their locations) and we assume that the initial values for  $w_l(\mathbf{s}_i, 0)$  follow the normal distribution with 0 and variance 0.5 independently for  $l = 1$  under this model.

In this study we illustrate throughout with the Metropolis-Hastings algorithm for simulating



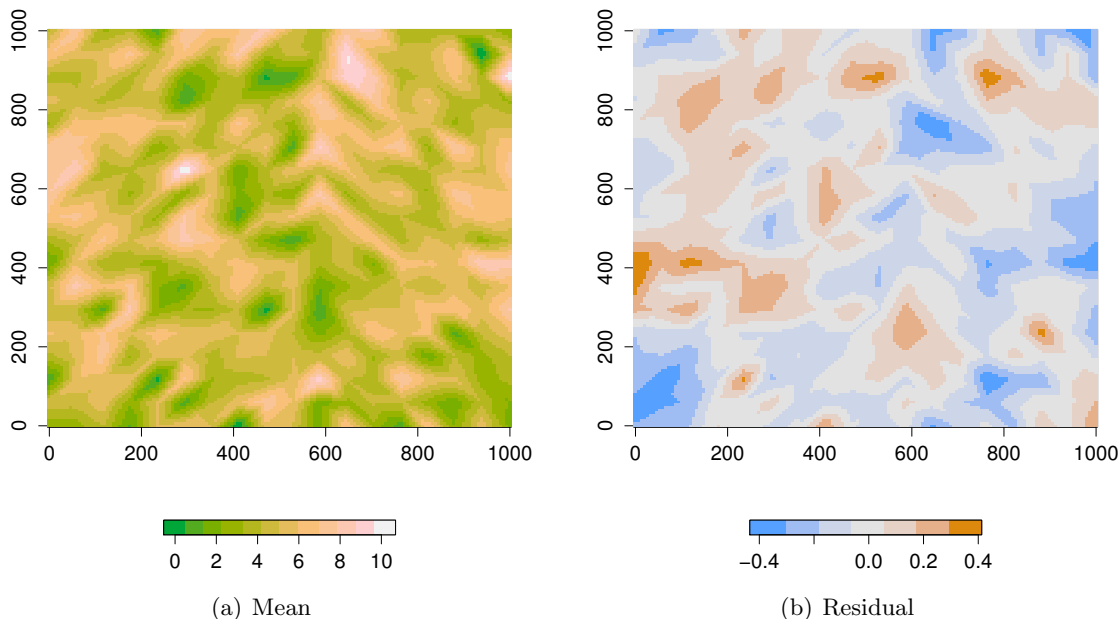


Figure 2: Fitted surface plots for (a) mean and corresponding (b) residuals using GPP-based model for time point five.

from the conditional distribution of the spatial decay parameter since that was found to be the best among all three methods available in the package (Bakar 2012). In all our MCMC implementations we use a small number of iterations to choose the tuning parameter (mostly by trial and error) that achieves an acceptance rate between 15 to 40%, see Section 2.6. For the simulation example, the Gibbs sampler is then run for 5,000 iterations for making inference with first 1,000 iterations as burn-in. We replicate the experiment 25 times and store the MCMC iterates of the model parameters, see Table 2 for the summary statistics.

The Gibbs sampler produced estimates of the parameters that were very close to the true simulation values, see Table 2. Moreover, the acceptance rate for sampling the decay parameter  $\phi$  was also reasonable.

### *Fitted plots and missing values*

Figure 2(a) shows the fitted mean surface plots for time point 5 with the GPP-based model. Note that, in this simulation example we only take 30 time points to reduce the computational burden of model fitting. The fitted surface plot for the mean is obtained using `spTimer` S3 class function `plot`, by using the argument `surface` inside the function. The argument `surface` is set to "Mean" or "SD" to plot the fitted mean or standard deviations respectively. It is also possible to change the color palette of the surface plot using the argument `col`; for example, we can use package `colorspace` (Zeileis, Hornik, and Murrell 2009) and write `col = rainbow_hcl(100, start = 200, end = 0)`. In addition, argument `a3d = TRUE` can be used to obtain a 3-dimensional plot of the fitted observations. Moreover, contours can be added to the plots. The accompanying R file, `v63i15.R` contains the detailed code for doing this by sending the fitted model output to two new functions `plot.spT` and `contour.spT` which are also provided. Note that these functions will require the `akima` (Akima 2013) package.

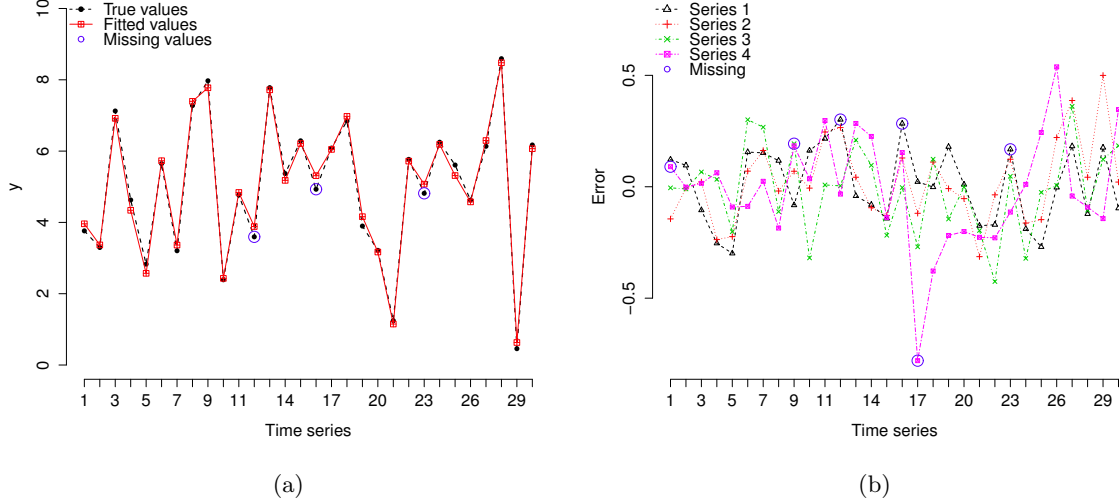


Figure 3: (a) Time series plots of the true and fitted values at one randomly chosen location. (b) Time series plots of the residuals at 4 randomly chosen locations.

In Figure 2(b) we provide a residual ( $z - \hat{z}$ ) surface plot. We observe that the residual surface plot varies from  $-0.4$  to  $0.4$  and in most places the fitted values are close to the true observations.

A time series plot of the fitted values is shown in Figure 3(a), for the first location in the simulated data set, using the GPP model. We see that the fitted mean is very close to the true value, and in addition it also shows that the estimates of the missing values are very close to the true value. We also provide residual plots for 4 randomly chosen locations in Figure 3(b), where we see the time series plots for the error are close to zero without any visible dependence structure. The missing values are indicated by circles, which are fitted using the Bayesian modeling discussed in Section 2.

### *Sensitivity of the signal-to-noise ratio (SNR)*

We also check the sensitivity of the fit for different signal-to-noise ratio (SNR), i.e.,  $\zeta = \sigma_\eta^2 / \sigma_\epsilon^2$ , through the simulated data. We use 3 different scenarios: (1)  $\sigma_\eta^2$  and  $\sigma_\epsilon^2$  are same, i.e.,  $\zeta = 1$ , (2) when  $\sigma_\eta^2$  is 10 times higher than  $\sigma_\epsilon^2$ , i.e.,  $\zeta = 10$  and (3) when  $\zeta = 15$ . We do not allow  $\zeta$  to be less than one since it is very unusual for the nugget effect,  $\sigma_\epsilon^2$ , to be larger than the spatial variance,  $\sigma_\eta^2$ .

We simulate data sets in 144 locations for 365 days for the intercept only model defined earlier in Section 4.1. We also replicate the procedure 25 times and obtain the MCMC results for the variance parameters. Figure 4 shows density plots of  $\zeta$  for the MCMC samples obtained from the GP model. The true value of  $\zeta$  is shown using a vertical line. We observe, the distribution of  $\zeta$ 's for different scenarios include its true value. The MCMC summary statistics for  $\beta_0$ , i.e., the intercept coefficient is given in Table 3. Note that the 95% credible intervals are almost totally unaffected by the value of  $\zeta$ .

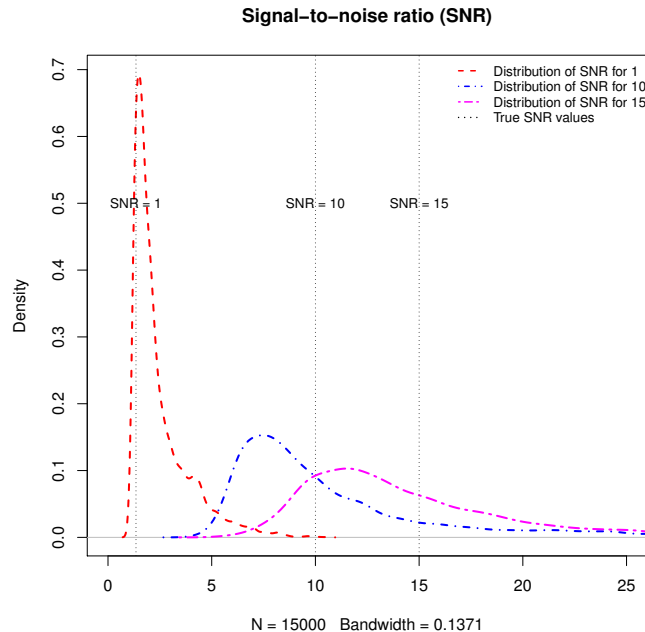


Figure 4: Distribution of signal-to-noise ratio (SNR),  $\zeta$ , for 3 different true values: 1, 10 and 15.

	True $\beta_0$	Median	95% Interval
$\zeta = 1$	5.0	5.01	(4.96, 5.07)
$\zeta = 10$	5.0	4.98	(4.94, 5.03)
$\zeta = 15$	5.0	5.02	(4.97, 5.08)

Table 3: Summary statistics for  $\beta_0$  for different scenarios of signal-to-noise ratio ( $\zeta$ ).

## 4.2. Comparison study

In this section we compare the performance of **spTimer** models with **spBayes** and also with a non-Bayesian model, e.g., the additive model as implemented in the **mgcv** package. From a modeling perspective we particularly focus on the issues of model fitting, prediction and computation time. To proceed, we start with a simple time series data having just 5 time points and then increase this to 10, 20 and 60. We simulate 49 locations (see Figure 5(a)) from the spatial domain defined in Section 4.1; where in addition to the grid points, the sampling locations are also chosen randomly, see Figure 5(b). We use the same model parameters as defined earlier in Section 4.1. Note that in this section we consider the intercept only model and obtain 4,000 MCMC samples after discarding first 1,000 samples as burn-in.

Figure 6 represents the density plots for the coefficients used in the models. As expected, we observe that all three packages provide estimates of the  $\beta_0$  parameter close to the true value. However, we also observe that the estimate obtained from **spTimer** has a shorter length credible interval than that using the **spBayes** package.

Table 4 shows a comparison of approximate computation times for the GP models using **spTimer**, and **spBayes** for data sets with different length of time series. Computation times for using the **mgcv** package are not included in this table since that package does not require

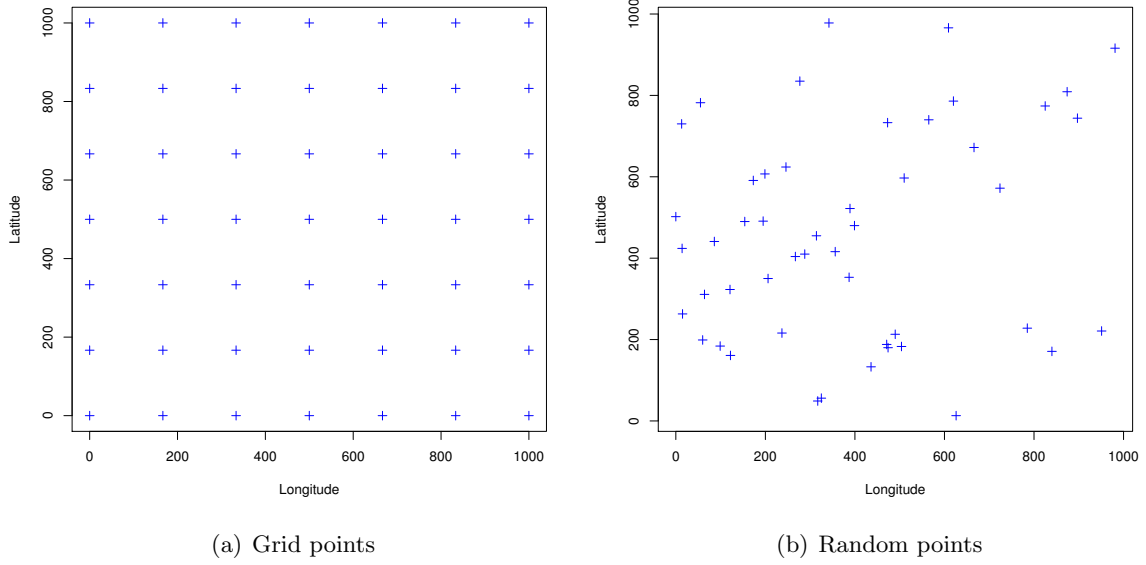


Figure 5: A representation of the 49 (a) grid and (b) random locations (+) used for simulating the data for comparison study.

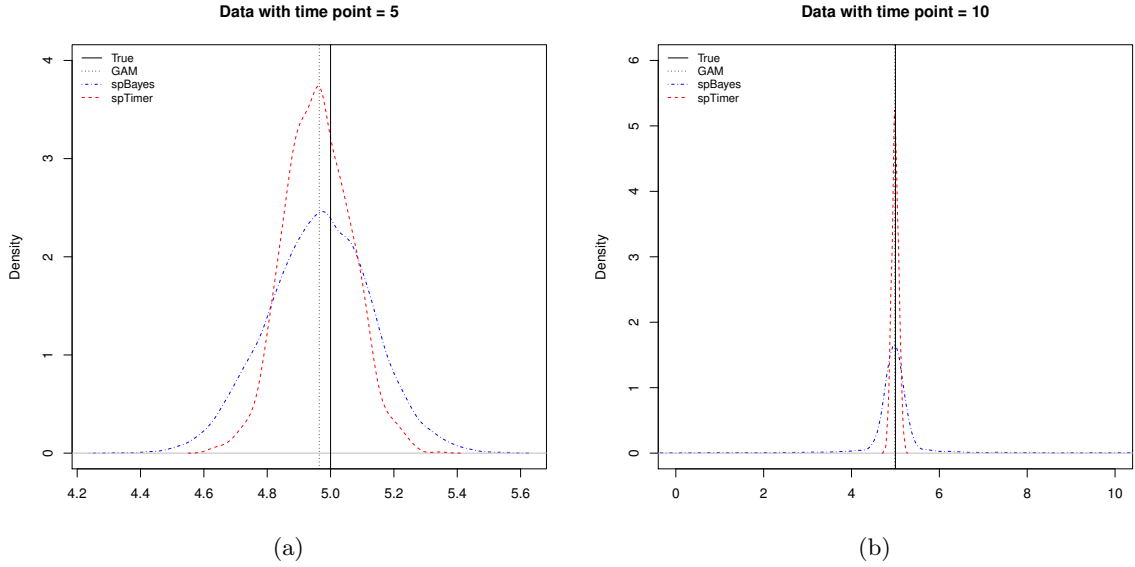


Figure 6: Density plots for the estimated  $\beta$  parameters obtained from the simulated data sets with time points (a) 5 and (b) 10. True value for  $\beta = 5$  is also superimposed using vertical solid line.

an iterative fitting method such as MCMC. This table clearly shows that **spTimer** provides faster model fitting than **spBayes** when the number of time points in the data is greater than one. Computation times for implementing the model when there is data for exactly one time point are comparable for the **spTimer** and **spBayes** packages and hence are not shown here.

To compare the off-site predictive performance we set aside 20% data for validation purposes. We use a small number of time points  $T = 5$  and a large number  $T = 60$  to compare

Models	Computation time			
	$T = 5$	$T = 10$	$T = 20$	$T = 60$
GP ( <b>spBayes</b> )	3 Minutes	25 Minutes	4 Hours	-
GP ( <b>spTimer</b> )	10 Seconds	12 Seconds	16 Seconds	40 Seconds

Table 4: Approximate computation time for model fitting using **spTimer** and **spBayes** using data sets simulated from different time series.

Models	$T = 5$		$T = 60$		
	MSE	MAE	MSE	MAE	
Grid-based	GP ( <b>spTimer</b> )	0.0898	0.2380	0.0759	0.2202
	GP ( <b>spBayes</b> )	1.0692	0.5918	-	-
	GAM ( <b>mgcv</b> )	0.1359	0.3021	0.1021	0.2555
Random	GP ( <b>spTimer</b> )	0.0970	0.2530	0.0680	0.2090
	GP ( <b>spBayes</b> )	1.0897	0.6306	-	-
	GAM ( <b>mgcv</b> )	0.1097	0.2689	0.1004	0.2579

Table 5: Comparison of validation statistics, mean squared error (MSE) and mean absolute error (MAE) obtained from **spTimer**, **spBayes** and **mgcv** models.

the models. We observe that both the validation criteria, MSE and MAE, are smallest for the **spTimer** model for the spatio-temporal data, see Table 5. Note that we omit the predictive results for **spBayes** at  $T = 60$ , because it is computationally prohibitive as it does not finish computations for predictions even after running continuously for several days, although the model fitting job finishes within a day as reported in Table 4. We also provide further comparison of predictive performance of the models using the real life data example in Section 5.

## 5. A practical example

We use a real life data set, previously analysed by [Sahu and Bakar \(2012a\)](#), on daily maximum 8-hour average ground level ozone concentration for the months of July and August in 2006, observed at 28 monitoring sites in the state of New York. We consider three important covariates: maximum temperature (**cMAXTEMP** in degree Celsius), wind speed (**WDSP** in nautical miles) and percentage average relative humidity (**RH**) for building a spatio-temporal model for ozone concentration. Further details regarding the covariate values and their spatial interpolation are provided in [Bakar \(2012\)](#). Figure 7 represents a map of the study region together with the 28 monitoring locations of which 8 have been set aside for model validation purposes. Moreover, we also set aside the data for the last 2 days (August 30 and 31) for validating the temporal forecasts. The following set of code lines are used for data preparation.

```
R> data("NYdata")
R> s <- c(8, 11, 12, 14, 18, 21, 24, 28)
R> DataFit <- spT.subset(data = NYdata, var.name = "s.index", s = s,
+   reverse = TRUE)
R> DataFit <- subset(DataFit,
+   with(DataFit, !(Day %in% c(30, 31) & Month == 8)))
```

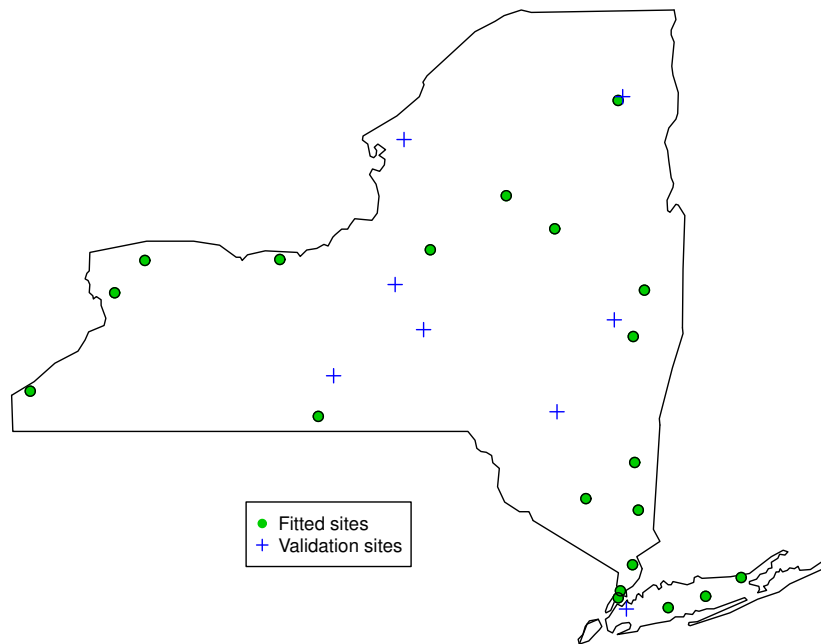


Figure 7: A map of the state of New York showing locations of the 28 ozone monitoring sites of which data from 8 are used for validation purposes.

```
R> DataValPred <- spT.subset(data = NYdata, var.name = "s.index", s = s)
R> DataValPred <- subset(DataValPred,
+   with(DataValPred, !(Day %in% c(30, 31) & Month == 8)))
```

where, `DataFit` is for model fitting, and the data sets `DataValPred` is for model validation.

To fit GP model using **spTimer** we use the following code:

```
R> set.seed(11)
R> post.gp <- spT.Gibbs(formula = o8hrmax ~ cMAXTMP + WDSP + RH,
+   data = DataFit, model = "GP",
+   coords = ~ Longitude + Latitude, scale.transform = "SQRT",
+   spatial.decay = spT.decay(distribution = Gamm(2, 1), tuning = 0.1))
```

A number of remarks are in order. The fitted model is the GP model, see Equations 2 and 3. The linear (covariate) part of the model is specified by the `formula` argument that automatically includes the intercept. Secondly, the square-root transformation is used, on the fly, to stabilize the variance (Sahu *et al.* 2007; Sahu and Bakar 2012a). The initial values and the values of the hyper-parameters for the prior distributions are assumed by default. Moreover, by default MCMC is run for 5,000 further iterations after discarding first 1,000. The **spTimer** user manual lists all the defaults and the ways to change them.

The package **spTimer** provides the usual R `print` and `summary` commands for obtaining summaries of the model fit. Here is some sample output:



```
R> print(post.gp)
```

```
-----
Model: GP
Call: o8hrmax ~ cMAXTMP + WDSP + RH
Iterations: 5000
nBurn: 1000
Acceptance rate for phi (%): 32.58
-----
```

```
          Goodness.of.fit Penalty   PMCC
values:          253.31  631.76 885.07
-----
```

```
Computation time: 6.38 - Sec.
```

We can also view the MCMC trace plots of the parameters using the `plot` function as:

```
R> plot(post.gp)
```

Figure 8 shows the MCMC trace plots for the four model parameters for 15000 iterations. Further MCMC diagnostics can be done using **coda** (Plummer, Best, Cowles, and Vines 2012). For example, one can use the code

```
R> autocorr.diag(as.mcmc(post.gp))
```

to generate the autocorrelation plots. One can also obtain residual plots using the function `plot` with the additional argument `residuals = TRUE` as:

```
R> plot(post.gp, residuals = TRUE)
```

However, none of these plots are included here for brevity.

The predictive model choice criteria (PMCC) described in Section 2.6 is obtained as `post.gp$PMCC`. We obtain the parameter estimates from the MCMC samples using the familiar `summary` command:

```
R> summary(post.gp)
```

```
-----
Model: GP
Call: o8hrmax ~ cMAXTMP + WDSP + RH
Iterations: 5000
nBurn: 1000
Acceptance rate for phi (%): 32.58
-----
```

```
          Goodness.of.fit Penalty   PMCC
values:          253.31  631.76 885.07
-----
```

```
Computation time: 6.38 - Sec.
```

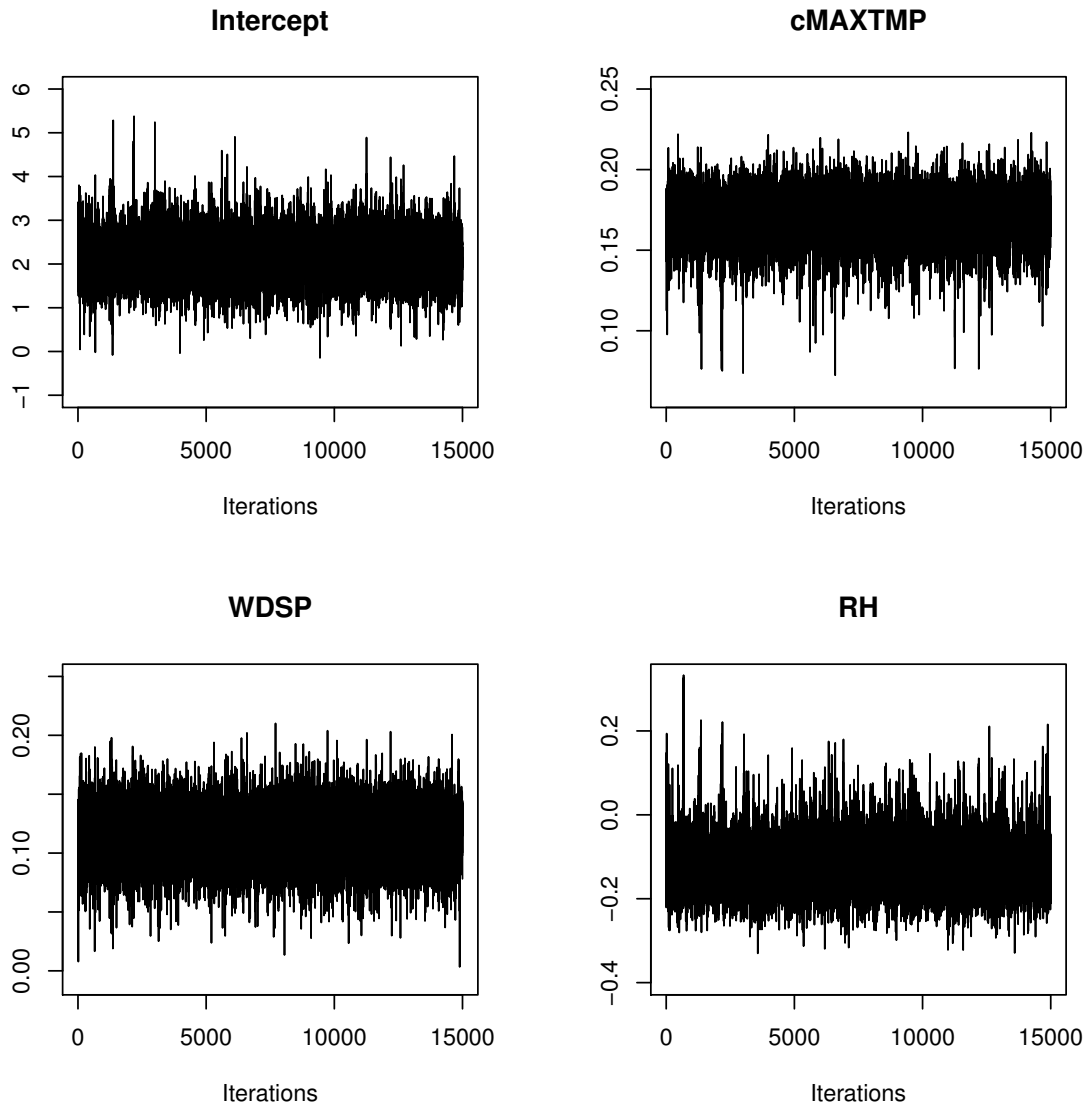


Figure 8: MCMC traceplots for four model parameters using GP model.

---

Parameters:

	Mean	Median	SD	Low2.5p	Up97.5p
(Intercept)	3.1790	3.1058	0.8474	1.7178	5.1722
cMAXTMP	0.1337	0.1374	0.0262	0.0685	0.1746
WDSP	0.0770	0.0787	0.0303	0.0134	0.1331
RH	-0.0544	-0.0655	0.0977	-0.2141	0.1752
sig2eps	0.0167	0.0160	0.0043	0.0105	0.0268
sig2eta	0.8305	0.6905	0.4747	0.4789	2.1902
phi	0.0069	0.0072	0.0023	0.0020	0.0108

---

These parameter estimates show that except for RH, all regression coefficients are statistically significant for the GP model since the 95% credible intervals do not contain zero. The estimate of the spatial decay parameter  $\phi = 0.007$  implies an effective range of 427 kilometers. We

also observe that, as expected, the spatial variance  $\sigma_{\eta}^2$  is higher than the nugget effect  $\sigma_{\epsilon}^2$ .

Prediction capabilities of **spTimer** are explored using the `predict` function, as noted in Table 1. After the above model fitting we can use the following code lines to perform and examine prediction.

```
R> set.seed(11)
R> pred.gp <- predict(post.gp, newdata = DataValPred,
+   newcoords = ~ Longitude + Latitude)
R> print(pred.gp)
```

```
-----
Spatial prediction with Model: GP
Covariance function: exponential
Distance method: geodetic:km
Computation time: 3.39 - Sec.
-----
```

For model validation **spTimer** has a built-in function for different validation criteria which have been discussed in Section 3.3. Hence, we write:

```
R> spT.validation(DataValPred$o8hrmax, c(pred.gp$Median))
```

```
##
Mean Squared Error (MSE)
Root Mean Squared Error (RMSE)
Mean Absolute Error (MAE)
Mean Absolute Percentage Error (MAPE)
Bias (BIAS)
Relative Bias (rBIAS)
Relative Mean Separation (rMSEP)
##
      MSE      RMSE      MAE      MAPE      BIAS      rBIAS      rMSEP
43.3210  6.5819  5.0803 11.8626  0.6350  0.0136  0.2621
```

The package **spTimer** can also perform temporal forecasting at both observed and unobserved locations using the same `predict` function when it is called with the additional argument `type = "temporal"` – details are provided in the package manual.

A predictive map can be drawn using the predictive output obtained from the `predict` function. Figure 9 provides such maps for daily ozone concentration levels and their standard deviations on August 29, 2006. The accompanying R file `v63i15.R` provides the code to produce these figures.

We conclude this example with a comparison study with the non-Bayesian generalized additive models (Hastie and Tibshirani 1990) using the R package **mgcv** (Wood 2006) as suggested by a reviewer. The GP models, with typical code as given above, are fitted to the data from the 20 fitting sites and then spatial predictions are obtained for the 8 set aside validation sites. The code for implementing this comparison study is provided in the accompanying R file, `v63i15.R`.

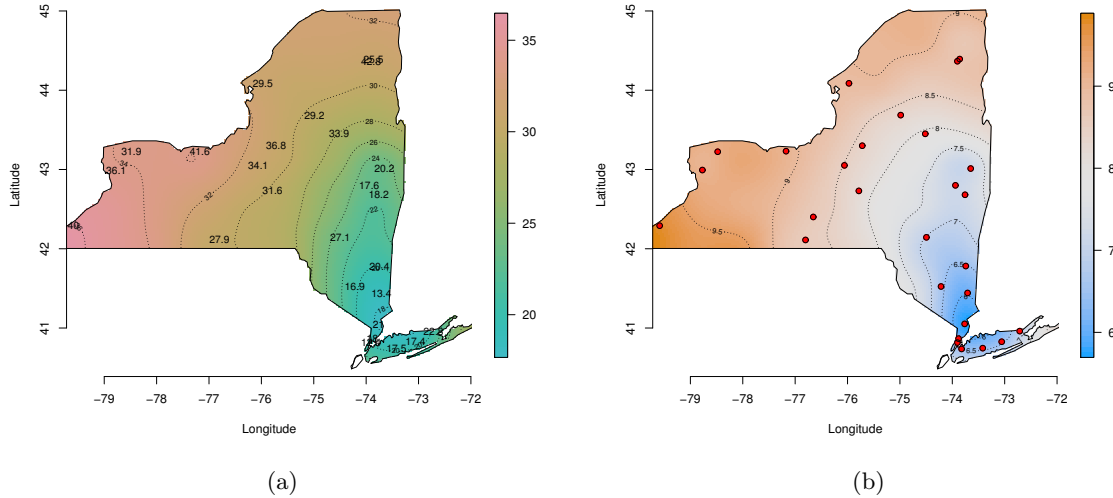


Figure 9: Spatially interpolated plots of the daily maximum 8-hour ozone concentration levels and (b) their standard deviations obtained from the GP models for 29 August, 2006. Actual observations and their locations are also superimposed in plots (a) and (b) respectively.

Models		MSE	MAE	rBIAS	rMSEP
Bayesian space-time	GP	43.32	5.08	0.01	0.26
Non-Bayesian	GAM	100.76	7.89	-0.03	0.60

Table 6: Validation statistics for the GP and additive models (GAM).

For the additive model we use the `gam` function and fit additive model and report the predictive performance of the model, see Table 6. We observe that the MSE for the Bayesian space-time GP model is reduced by about 56% compared to the generalized additive models, showing superiority of the spatio-temporal models implemented in the package **spTimer**. Further similar comparison studies have been performed by Crimp, Bakar, Kokic, Jin, Nicholls, and Howden (2015), where **spTimer** GP model also shows better predictive performance compared to the additive models for analyzing frost levels in south-east Australia.

## 6. Summary

This paper introduces the contributed R package **spTimer** that enables model fitting, spatial and temporal predictions for large structured point-referenced spatio-temporal data sets. Currently, the package is able to analyze data using three substantial and well established spatio-temporal models. The package also includes a number of attractive features ranging from on the fly transformation to the ability to infer for certain temporal aggregates. The main body of the code has been validated using a substantial simulation example and a real life data example.

The underlying code for the package has been written using the C programming language that is very portable across many different operating systems such as Microsoft Windows, Linux and Macintosh. The end-user, however, does not need to work with the C language as all the

analysis can be performed by using commands in R. The MCMC-based Bayesian hierarchical modeling, as implemented in the package, is relatively fast for moderate (a few thousand) to large (more than a million) data sets. In particular, the GPP-based model is the fastest to run as we have reported in our related work (Sahu and Bakar 2012b).

The package can be extended in several ways, for example, for modeling multivariate data and for modeling data with a non-Gaussian first stage model. In addition, it will be very fruitful to add modeling capabilities for spatially varying coefficient process models. Other possible extensions include the ability to handle data from sensor networks that vary over time. Moreover, the implemented models can be enhanced to model mixture of discrete and continuous data such as rainfall. Lastly, extension of the package for handling spatially mis-aligned data will also be of considerable interest in the literature.

## Acknowledgments

The authors thank Dr. Philip Kokic, Dr. Huidong Jin, Dr. Sabyasachi Mukhopadhyay, Mr. Mark Bass, and two anonymous reviewers for their many valuable comments on the package and on previous drafts of this paper.

## References

- Akima H (2013). *akima: Interpolation of Irregularly Space Data*. R package version 0.5-11, URL <http://CRAN.R-project.org/package=akima>.
- Baddeley A, Turner R (2005). “**spatstat**: An R Package for Analyzing Spatial Point Patterns.” *Journal of Statistical Software*, **12**(6), 1–42. URL <http://www.jstatsoft.org/v12/i06/>.
- Bakar KS (2012). *Bayesian Analysis of Daily Maximum Ozone Levels*. PhD Thesis, University of Southampton, Southampton.
- Banerjee S, Carlin BP, Gelfand AE (2004). *Hierarchical Modeling and Analysis for Spatial Data*. Chapman & Hall/CRC, Boca Raton.
- Banerjee S, Gelfand AE, Finley AO, Sang H (2008). “Gaussian Predictive Process Models for Large Spatial Data Sets.” *Journal of the Royal Statistical Society B*, **70**, 825–848.
- Bivand RS, Pebesma E, Gómez-Rubio V (2013). *Applied Spatial Data Analysis with R*. 2nd edition. Springer-Verlag, New York. URL <http://www.asdar-book.org/>.
- Cameletti M, Ignaccolo R, Bande S (2009). “Comparing Air Quality Statistical Models.” *Technical report*, University of Bergamo, Italy.
- Christensen OF, Ribeiro PJ (2002). “**geoRglm**: A Package for Generalised Linear Spatial Models.” *proglangR News*, **2**(2), 26–28. URL <http://CRAN.R-project.org/doc/Rnews/>.
- Cressie NAC (1993). *Statistics for Spatial Data*. John Wiley & Sons, New York.
- Cressie NAC, Wikle CK (2011). *Statistics for Spatio-Temporal Data*. John Wiley & Sons, New York.

- Crimp S, Bakar KS, Kokic P, Jin H, Nicholls N, Howden M (2015). “Bayesian Space-Time Model to Analyse Frost Risk for Agriculture in Southeast Australia.” *International Journal of Climatology*. doi:10.1002/joc.4109.
- de los Campos G, Perez Rodriguez P (2014). *BLR: Bayesian Linear Regression*. R package version 1.4, URL <http://CRAN.R-project.org/package=BLR>.
- Diggle PJ, Ribeiro PJ (2007). *Model-Based Geostatistics*. Springer-Verlag, New York.
- Finley AO, Banerjee S, Carlin BP (2007). “**spBayes**: An R Package for Univariate and Multivariate Hierarchical Point-Referenced Spatial Models.” *Journal of Statistical Software*, **19**(4), 1–24. URL <http://www.jstatsoft.org/v19/i04/>.
- Finley AO, Banerjee S, Gelfand AE (2015). “**spBayes** for Large Univariate and Multivariate Point-Referenced Spatio-Temporal Data Models.” *Journal of Statistical Software*, **63**(13), 1–28. URL <http://www.jstatsoft.org/v63/i13/>.
- Finley AO, Sang H, Banerjee S, Gelfand AE (2009). “Improving the Performance of Predictive Process Modeling for Large Datasets.” *Computational Statistics & Data Analysis*, **53**, 2873–2884.
- Gelfand AE (2012). “Hierarchical Modeling for Spatial Data Problems.” *Spatial Statistics*, **1**, 30–39.
- Gelfand AE, Diggle PJ, Fuentes M, Guttorp P (2010). *The Handbook of Spatial Statistics*. Chapman & Hall/CRC, New York.
- Gelfand AE, Ghosh SK (1998). “Model Choice: A Minimum Posterior Predictive Loss Approach.” *Biometrika*, **85**, 1–11.
- Gelfand AE, Smith AFM (1990). “Sampling-Based Approaches to Calculating Marginal Densities.” *Journal of the American Statistical Association*, **85**(410), 398–409.
- Gelman A, Carlin JB, Stern HS, Rubin DB (2004). *Bayesian Data Analysis*. 2nd edition. Chapman & Hall/CRC, Boca Raton.
- Hadfield JD (2010). “MCMC Methods for Multi-Response Generalized Linear Mixed Models: The **MCMCglmm** R Package.” *Journal of Statistical Software*, **33**(2), 1–22. URL <http://www.jstatsoft.org/v33/i02/>.
- Handcock MS, Stein ML (1993). “A Bayesian Analysis of Kriging.” *Technometrics*, **35**, 403–410.
- Handcock MS, Wallis J (1994). “An Approach to Statistical Spatial-Temporal Modelling of Meteorological Fields.” *Journal of the American Statistical Association*, **89**, 368–390.
- Hastie T (2014). *gam: Generalized Additive Models*. R package version 1.09.1, URL <http://CRAN.R-project.org/package=gam>.
- Hastie T, Tibshirani R (1990). *Generalized Additive Models*. Chapman & Hall, London.



- Krige DG (1951). “A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand.” *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, **52**, 119–139.
- Martin AD, Quinn KM, Park JH (2011). “**MCMCpack**: Markov Chain Monte Carlo in R.” *Journal of Statistical Software*, **42**(9), 1–21. URL <http://www.jstatsoft.org/v42/i09/>.
- Matérn B (1986). *Spatial Variation*. 2nd edition. Springer-Verlag, Berlin.
- Millo G, Piras G (2012). “**splm**: Spatial Panel Data Models in R.” *Journal of Statistical Software*, **47**(1), 1–38. URL <http://www.jstatsoft.org/v47/i01/>.
- Nychka D, Furrer R, Sain S (2014). **fields**: *Tools for Spatial Data*. R package version 7.1, URL <http://CRAN.R-project.org/package=fields>.
- Pebesma E (2012). “**spacetime**: Spatio-Temporal Data in R.” *Journal of Statistical Software*, **51**(7), 1–30. URL <http://www.jstatsoft.org/v51/i07/>.
- Pebesma EJ (2004). “Multivariable Geostatistics in S: The **gstat** Package.” *Computers and Geosciences*, **30**, 683–691.
- Pebesma EJ, Bivand RS (2005). “Classes and Methods for Spatial Data in R.” *R News*, **5**(2), 9–13. URL <http://CRAN.R-project.org/doc/Rnews/>.
- Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2014). **nlme**: *Linear and Nonlinear Mixed Effects Models*. R package version 3.1-118, URL <http://CRAN.R-project.org/package=nlme>.
- Plummer M (2014). **rjags**: *Bayesian Graphical Models Using MCMC*. R package version 3-14, URL <http://CRAN.R-project.org/package=rjags>.
- Plummer M, Best N, Cowles K, Vines K (2006). “**coda**: Convergence Diagnosis and Output Analysis for MCMC.” *R News*, **6**(1), 7–11. URL <http://CRAN.R-project.org/doc/Rnews/>.
- Plummer M, Best N, Cowles K, Vines K (2012). **coda**: *Output Analysis and Diagnostics for MCMC*. R package version 0.16-1, URL <http://CRAN.R-project.org/package=coda>.
- R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.
- Ribeiro PJ, Diggle PJ (2001). “**geoR**: A Package for Geostatistical Analysis.” *proglangR News*, **1**(2), 14–18. URL <http://CRAN.R-project.org/doc/Rnews/>.
- Ripley B (2013). **spatial**: *Functions for Kriging and Point Pattern Analysis*. R package version 7.3-7, URL <http://CRAN.R-project.org/package=spatial>.
- Sahu SK, Bakar KS (2012a). “A Comparison of Bayesian Models for Daily Ozone Concentration Levels.” *Statistical Methodology*, **9**(1), 144–157.
- Sahu SK, Bakar KS (2012b). “Hierarchical Bayesian Autoregressive Models for Large Space Time Data with Applications to Ozone Concentration Modelling.” *Applied Stochastic Models in Business and Industry*, **28**, 395–415.

- Sahu SK, Gelfand AE, Holland DM (2006). “Spatio-Temporal Modeling of Fine Particulate Matter.” *Journal of Agricultural, Biological, and Environmental Statistics*, **11**, 61–86.
- Sahu SK, Gelfand AE, Holland DM (2007). “High-Resolution Space-Time Ozone Modeling for Assessing Trends.” *Journal of the American Statistical Association*, **102**, 1221–1234.
- Sahu SK, Gelfand AE, Holland DM (2010). “Fusing Point and Areal Level Space-Time Data with Application to Wet Deposition.” *Journal of the Royal Statistical Society C*, **59**, 77–103.
- Spiegelhalter DJ, Thomas A, Best NG, Lunn D (2003). *WinBUGS Version 1.4 User Manual*. MRC Biostatistics Unit, Cambridge. URL <http://www.mrc-bsu.cam.ac.uk/bugs/>.
- Stein ML (1999). *Statistical Interpolation of Spatial Data: Some Theory for Kriging*. Springer-Verlag, New York.
- Sturtz S, Ligges U, Gelman A (2005). “**R2WinBUGS**: A Package for Running **WinBUGS** from R.” *Journal of Statistical Software*, **12**(3), 1–16. URL <http://www.jstatsoft.org/v12/i03/>.
- Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York.
- Wood SN (2006). *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC.
- Zeileis A, Hornik K, Murrell P (2009). “Escaping RGBland: Selecting Colors for Statistical Graphics.” *Computational Statistics & Data Analysis*, **53**, 3259–3270.
- Zhang H (2004). “Inconsistent Estimation and Asymptotically Equal Interpolations in Model-Based Geostatistics.” *Journal of the American Statistical Association*, **99**, 250–261.

## A. Full conditional distributions for the GP model

- The full conditional distribution of  $\boldsymbol{\beta}$  can be obtained from the kernel of (4) as:  $\pi(\boldsymbol{\beta}|\dots, \mathbf{z}) \sim N(\Delta\boldsymbol{\chi}, \Delta)$ , where

$$\Delta^{-1} = \sum_{l=1}^r \sum_{t=1}^{T_l} \mathbf{X}_{lt}^\top \Sigma_\eta^{-1} \mathbf{X}_{lt} + \mathbf{I}_p / \delta_\beta^2$$

$$\boldsymbol{\chi} = \sum_{l=1}^r \sum_{t=1}^{T_l} \mathbf{X}_{lt}^\top \Sigma_\eta^{-1} \mathbf{O}_{lt}.$$

- Similarly from (4), we sample  $\sigma_\epsilon^2$  and  $\sigma_\eta^2$  from the following conditional distributions respectively:

$$\pi(1/\sigma_\epsilon^2|\dots, \mathbf{z}) \sim G\left(\frac{N}{2} + a, b + \frac{1}{2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{Z}_{lt} - \mathbf{O}_{lt})^\top (\mathbf{Z}_{lt} - \mathbf{O}_{lt})\right),$$

$$\pi(1/\sigma_\eta^2|\dots, \mathbf{z}) \sim G\left(\frac{N}{2} + a, b + \frac{1}{2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta})^\top S_\eta^{-1} (\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta})\right).$$

- From the kernel of the joint density (4), we obtain the full conditional distribution for  $\mathbf{O}_{lt}$  as  $\pi(\mathbf{O}_{lt}|\dots, \mathbf{z}) \sim N(\Delta_{lt}\boldsymbol{\chi}_{lt}, \Delta_{lt})$ , where:

$$\Delta_{lt}^{-1} = \mathbf{I}_n / \sigma_\epsilon^2 + \Sigma_\eta^{-1}$$

$$\boldsymbol{\chi}_{lt} = \mathbf{Z}_{lt} / \sigma_\epsilon^2 + \Sigma_\eta^{-1} \mathbf{X}_{lt} \boldsymbol{\beta}.$$

- The full conditional distribution of  $\phi$  is non-standard and is given by:

$$\pi(\phi|\dots, \mathbf{z}) \propto \pi(\phi) \times |\mathbf{S}_\eta|^{-\sum_{l=1}^r T_l/2} \times \exp\left[-\frac{1}{2\sigma_\eta^2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta})^\top S_\eta^{-1} (\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta})\right].$$

## B. Full conditional distributions for the AR model

- The full conditional distribution of  $\boldsymbol{\beta}$  can be obtained from the joint posterior distribution of AR models (7) as:  $\pi(\boldsymbol{\beta}|\dots, \mathbf{z}) \sim N(\Delta\boldsymbol{\chi}, \Delta)$ , where:

$$\Delta^{-1} = \sum_{l=1}^r \sum_{t=1}^{T_l} \mathbf{X}_{lt}^\top \Sigma_\eta^{-1} \mathbf{X}_{lt} + \mathbf{I}_p / \delta_\beta^2$$

$$\boldsymbol{\chi} = \sum_{l=1}^r \sum_{t=1}^{T_l} \mathbf{X}_{lt}^\top \Sigma_\eta^{-1} (\mathbf{O}_{lt} - \rho \mathbf{O}_{lt}).$$

- The full conditional distribution of  $\rho$  can be obtained from (7) as:  $\pi(\rho|\dots, \mathbf{z}) \sim N(\Delta\chi, \Delta)$ , where:

$$\Delta^{-1} = \sum_{l=1}^r \sum_{t=1}^{T_l} \mathbf{O}_{lt}^\top \Sigma_\eta^{-1} \mathbf{O}_{lt} + \mathbf{I}_p / \delta_\rho^2$$

$$\chi = \sum_{l=1}^r \sum_{t=1}^{T_l} \mathbf{O}_{lt}^\top \Sigma_\eta^{-1} (\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta}).$$

- For  $\sigma_\epsilon^2$  and  $\sigma_\eta^2$  we sample from the following conditional distributions respectively:

$$\pi(1/\sigma_\epsilon^2|\dots, \mathbf{z}) \sim G\left(\frac{N}{2} + a, b + \frac{1}{2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{Z}_{lt} - \mathbf{O}_{lt})^\top (\mathbf{Z}_{lt} - \mathbf{O}_{lt})\right),$$

$$\pi(1/\sigma_\eta^2|\dots, \mathbf{z}) \sim G\left(\frac{N}{2} + a, b + \frac{1}{2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{O}_{lt} - \rho\mathbf{O}_{lt-1} - \mathbf{X}_{lt}\boldsymbol{\beta})^\top S_\eta^{-1} (\mathbf{O}_{lt} - \rho\mathbf{O}_{lt-1} - \mathbf{X}_{lt}\boldsymbol{\beta})\right).$$

- From the joint posterior (7), we obtain the full conditional distribution for  $\mathbf{O}_{lt}$  for two cases: (1) when  $1 \leq t \leq T_l - 1$  and (2) when  $t = T_l$ . Hence, we write  $\pi(\mathbf{O}_{lt}|\dots, \mathbf{z}) \sim N(\Delta_{lt}\chi_{lt}, \Delta_{lt})$ , where:

**Case 1:**

$$\Delta_{lt}^{-1} = \mathbf{I}_n / \sigma_\epsilon^2 + (1 + \rho^2)\Sigma_\eta^{-1}$$

$$\chi_{lt} = \mathbf{Z}_{lt} / \sigma_\epsilon^2 + \Sigma_\eta^{-1} (\rho\mathbf{O}_{lt-1} + \mathbf{X}_{lt}\boldsymbol{\beta} + \rho(\mathbf{O}_{lt+1} - \mathbf{X}_{lt+1}\boldsymbol{\beta})).$$

**Case 2:**

$$\Delta_{lt}^{-1} = \mathbf{I}_n / \sigma_\epsilon^2 + \Sigma_\eta^{-1}$$

$$\chi_{lt} = \mathbf{Z}_{lt} / \sigma_\epsilon^2 + \Sigma_\eta^{-1} (\rho\mathbf{O}_{lt-1} + \mathbf{X}_{lt}\boldsymbol{\beta}).$$

- The full conditional distribution for  $\mathbf{O}_{l0}$  is  $N(\Delta_l\chi_l, \Delta_l)$ ,  $l = 1, \dots, r$  where

$$\Delta_l^{-1} = \rho^2 \Sigma_\eta^{-1} + \frac{1}{\sigma_l^2} S_0^{-1}$$

$$\chi_l = \rho(\mathbf{O}_{l1} - \mathbf{X}_{l1}\boldsymbol{\beta})^\top \Sigma_\eta^{-1} + \frac{1}{\sigma_l^2} \boldsymbol{\mu}_l^\top S_0^{-1}.$$

- We write the full conditional distribution for  $\boldsymbol{\mu}_l$  as  $N(\Delta_l\chi_l, \Delta_l)$ ,  $l = 1, \dots, r$ , where

$$\Delta_l^{-1} = \frac{1}{\sigma_l^2} S_0^{-1} + \frac{1}{\sigma_\mu^2} \mathbf{I}_n.$$

$$\chi_l = \frac{1}{\sigma_l^2} S_0^{-1} \mathbf{O}_{l0}$$

- The full conditional distribution for  $\sigma_l^2$

$$\pi(1/\sigma_l^2|\dots, \mathbf{z}) \sim G\left(\frac{n}{2} + a, b + \frac{1}{2} (\mathbf{O}_{l0} - \boldsymbol{\mu}_l)^\top S_0^{-1} (\mathbf{O}_{l0} - \boldsymbol{\mu}_l)\right), \quad l = 1, \dots, r.$$

- The full conditional distribution of  $\phi$  parameter is obtained from the kernel (7) as:

$$\begin{aligned} \pi(\phi|\dots, \mathbf{z}) &\propto \pi(\phi) \times |\mathbf{S}_\eta|^{-\sum_{l=1}^r T_l/2} \times \\ &\exp \left[ -\frac{1}{2\sigma_\eta^2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{O}_{lt} - \rho \mathbf{O}_{lt-1} - \mathbf{X}_{lt}\boldsymbol{\beta})^\top S_\eta^{-1} (\mathbf{O}_{lt} - \rho \mathbf{O}_{lt-1} - \mathbf{X}_{lt}\boldsymbol{\beta}) \right] \\ &\times |\mathbf{S}_0|^{-r/2} \times \exp \left[ -\frac{1}{2} \sum_{l=1}^r \frac{1}{\sigma_l^2} (\mathbf{O}_{l0} - \boldsymbol{\mu}_l)^\top S_0^{-1} (\mathbf{O}_{l0} - \boldsymbol{\mu}_l) \right]. \end{aligned}$$

## C. Full conditional distributions for the GPP-based AR model

The joint posterior distribution (10) is used to derive the full conditional distributions listed below.

- The full conditional distribution of  $\boldsymbol{\beta}$  is  $N(\Delta\chi, \Delta)$  where,

$$\begin{aligned} \Delta^{-1} &= \frac{1}{\sigma_\epsilon^2} \sum_{l=1}^r \sum_{t=1}^{T_l} X_{lt}^\top X_{lt} + \mathbf{I}_p / \delta_\beta^2, \\ \chi &= \frac{1}{\sigma_\epsilon^2} \sum_{l=1}^r \sum_{t=1}^{T_l} X_{lt}^\top (\mathbf{Z}_{lt} - A\mathbf{w}_{lt}). \end{aligned}$$

- The full conditional distribution of  $\rho$  is  $N(\Delta\chi, \Delta)I(0 < \rho < 1)$  where,

$$\begin{aligned} \Delta^{-1} &= \sum_{l=1}^r \sum_{t=1}^{T_l} \mathbf{w}_{lt-1}^\top \Sigma_\eta^{-1} \mathbf{w}_{lt-1} + \mathbf{I}_p / \delta_\rho^2 \\ \chi &= \sum_{l=1}^r \sum_{t=1}^{T_l} \mathbf{w}_{lt-1}^\top \Sigma_\eta^{-1} \mathbf{w}_{lt}. \end{aligned}$$

- The variance parameters  $\sigma_\epsilon^2$ ,  $\sigma_\eta^2$  and  $\sigma_l^2$  are sampled from:

$$\begin{aligned} \pi(1/\sigma_\epsilon^2|\dots, \mathbf{z}) &\sim G \left( \frac{N}{2} + a, b + \frac{1}{2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{Z}_{lt} - X_{lt}\boldsymbol{\beta} - A\mathbf{w}_{lt})^\top (\mathbf{Z}_{lt} - X_{lt}\boldsymbol{\beta} - A\mathbf{w}_{lt}) \right) \\ \pi(1/\sigma_\eta^2|\dots, \mathbf{z}) &\sim G \left( \frac{m \sum_{l=1}^r T_l}{2} + a, b + \frac{1}{2} \sum_{l=1}^r \sum_{t=1}^{T_l} (\mathbf{w}_{lt} - \rho \mathbf{w}_{lt-1})^\top \Sigma_\eta^{-1} (\mathbf{w}_{lt} - \rho \mathbf{w}_{lt-1}) \right) \\ \pi(1/\sigma_l^2|\dots, \mathbf{z}) &\sim G \left( \frac{m}{2} + a, b + \frac{1}{2} \mathbf{w}_{l0}^\top S_0^{-1} \mathbf{w}_{l0} \right), \quad l = 1, \dots, r. \end{aligned}$$

- The full conditional distribution of  $\mathbf{w}_{lt}$  is given by:  $N(\Delta_{lt}\chi_{lt}, \Delta_{lt})$  where

$$\Delta_{lt}^{-1} = \frac{1}{\sigma_\epsilon^2} A^\top A + (1 + \rho^2) \Sigma_\eta^{-1}$$

$$\chi_{lt} = \frac{1}{\sigma_\epsilon^2} A^\top (\mathbf{Z}_{lt} - X_{lt}\boldsymbol{\beta}) + \rho \Sigma_\eta^{-1} (\mathbf{w}_{lt-1} + \mathbf{w}_{lt+1}),$$

for  $1 \leq t \leq T_l - 1$ . For  $t = T_l$ , we have

$$\Delta_{lt}^{-1} = \frac{1}{\sigma_\epsilon^2} A^\top A + \Sigma_\eta^{-1}$$

$$\chi_{lt} = \frac{1}{\sigma_\epsilon^2} A^\top (\mathbf{Z}_{lt} - X_{lt}\boldsymbol{\beta}) + \rho \Sigma_\eta^{-1} \mathbf{w}_{lt-1}.$$

- Full conditional distribution for  $\mathbf{w}_{l0}$  is  $N(\Delta_l \chi_l, \Delta_l)$ ,  $l = 1, \dots, r$  where

$$\Delta_l^{-1} = \rho^2 \Sigma_\eta^{-1} + \frac{1}{\sigma_l^2} S_0^{-1}$$

$$\chi_l = \rho \Sigma_\eta^{-1} \mathbf{w}_{l1}.$$

- The full conditional distribution is not available in closed form and it is proportional to the joint posterior distribution provided in (10), Section 2.4.

### **Affiliation:**

Khandoker Shuvo Bakar  
 Department of Statistics  
 Yale University  
 New Haven, CT, 06511, United States of America  
 E-mail: [Shuvo.Bakar@yale.edu](mailto:Shuvo.Bakar@yale.edu)  
 URL: <http://statistics.yale.edu/>

Sujit K. Sahu  
 Southampton Statistical Sciences Research Institute  
 University of Southampton  
 Southampton, SO17 1BJ, United Kingdom  
 E-mail: [S.K.Sahu@soton.ac.uk](mailto:S.K.Sahu@soton.ac.uk)  
 URL: <http://www.soton.ac.uk/~sks/>