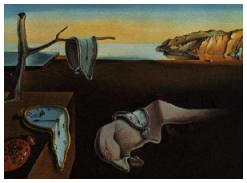


Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

spTimer: Spatio-Temporal Bayesian Modeling Using R

K.S. Bakar
CSIRO-MIS, Australia

S.K. Sahu
University of Southampton, UK

Abstract

Hierarchical Bayesian modeling of large point referenced space-time data are 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 analysis methods among practitioners and, hence there is an urgent need to develop high level software packages 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 uncer-

tainties in analysis, 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 Development Core Team 2006) packages are available for modeling and analyzing spatial data, e.g., the packages **sp** (Pebesma, Bivand, Rowlingson, and Rubio 2012a) and **spacetime** (Pebesma, Graeler, Gottfried, and Hijmans 2012b). Furthermore, packages **gstat** (Pebesma 2013), **splm** (Millo and Gianfranco 2012), **fields** (Furrer, Nychka, and Sain 2012) and **nlme** (Pinheiro, Bates, DebRoy, and Sarkar 2013) 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 2012) and **spatstat** (Baddeley and Turner 2012). However, none of these packages implement MCMC based Bayesian modeling methods.

General purpose R software packages, such as **MCMCpack** (Martin, Quinn, and Park 2012), **MCMCglmm** (Hadfield 2012), and **BLR** (Campos and Rodriguez 2012), 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 2013, 2007), **geoR** (Ribeiro and Diggle 2012, 2001), **geoRglm** (Christensen and Ribeiro Jr 2011; Christensen and Ribeiro 2002), **rjags** (Plummer 2012) and **R2WinBUGS** (Liggrs 2011) which is an R interface to WinBUGS (Spiegelhalter, Thomas, and Best 1999). 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 2011) and **mgcv** (Wood 2012) 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 Section 5.

The main contribution of this paper is the development of the R package **spTimer** that is available from the Comprehensive R Archive Network (CRAN) and is, henceforth, cited as (Bakar and Sahu 2013). 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 the 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 package 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 2012, 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) 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. A companion paper by Sahu, Bakar, and Awang (2012) illustrates the forecasting ability of all three models using a large space-time data set and the forecasting methods.

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:

$$\begin{aligned} \text{First} \quad & [data|process, parameter] \\ \text{Second} \quad & [process|parameter] \\ \text{Third} \quad & [parameter] \end{aligned}$$

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, $l = 1, \ldots, r$, and t denotes the shorter unit, e.g., day, $t = 1, \ldots, T$ where r and T 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 $\mathbf{s}_i$, $i = 1, ..., n$ at time denoted by two indices l and t. Let $\mathbf{Z}_{lt} = (Z_l(\mathbf{s}_1, t), ..., Z_l(\mathbf{s}_n, t))'$ and $\mathbf{O}_{lt} = (O_l(\mathbf{s}_1, t), ..., O_l(\mathbf{s}_n, t))'$. We shall denote all the observed data by $\mathbf{z}$ and $\mathbf{z}^*$ will denote all the missing data. Let $N = nrT$ be the total number of observations to be modeled.

Throughout, the notation $\boldsymbol{\epsilon}_{lt} = (\epsilon_l(\mathbf{s}_1, t), ..., \epsilon_l(\mathbf{s}_n, t))'$ will be used to denote the so called nugget effect or the pure error term assumed to be independently normally distributed $N(\mathbf{0}, \sigma^2_\epsilon \mathbf{I}_n)$ where σ^2_ϵ 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 $\boldsymbol{\eta}_{lt} = (\eta_l(\mathbf{s}_1, t), ..., \eta_l(\mathbf{s}_n, t))'$ and these will be assumed to follow $N(\mathbf{0}, \Sigma_\eta)$ independently in time, where $\Sigma_\eta = \sigma^2_\eta S_\eta$, σ^2_η is the site invariant spatial variance and S_η 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:

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

where $\Gamma(\nu)$ is the standard gamma function, K_ν is the modified Bessel function of second kind with order ν, and $||\mathbf{s}_i - \mathbf{s}_j||$ is the distance between sites $\mathbf{s}_i$ and $\mathbf{s}_j$. The parameter ϕ controls the rate of decay of the correlation as the distance $||\mathbf{s}_i - \mathbf{s}_j||$ increases and the parameter ν controls smoothness of the random field (Banerjee *et al.* 2004; Cressie 1993). The package **spTimer** allows several possibilities regarding estimation of the correlation parameters ϕ and ν 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 $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)$ 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:

$$\begin{aligned} \mathbf{Z}_{lt} &= \mathbf{O}_{lt} + \boldsymbol{\epsilon}_{lt}, & (2)\\ \mathbf{O}_{lt} &= \mathbf{X}_{lt}\boldsymbol{\beta} + \boldsymbol{\eta}_{lt}, & (3) \end{aligned}$$

for each $l = 1, \ldots, r$ and $t = 1, \ldots, T$, 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, $\mathbf{O}_{lt}$, $l = 1, \ldots, r$ and $t = 1, \ldots, T$. Let $\boldsymbol{\theta} = (\mathbf{O}, \boldsymbol{\beta}, \sigma^2_\epsilon, \sigma^2_\eta, \phi)$ 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:

$$\begin{aligned} \log \pi(\boldsymbol{\theta}, \mathbf{z}^*|\mathbf{z}) \quad \propto \quad & -\frac{N}{2}\log\sigma^2_\epsilon - \frac{1}{2\sigma^2_\epsilon}\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{Z}_{lt} - \mathbf{O}_{lt})'(\mathbf{Z}_{lt} - \mathbf{O}_{lt}) - \frac{N}{2}\log|\sigma^2_\eta S_\eta| \\ & -\frac{1}{2\sigma^2_\eta}\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta})' S^{-1}_\eta (\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta}) + \log\pi(\boldsymbol{\theta}). \end{aligned} \tag{4}$$

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:

$$\begin{aligned} \mathbf{Z}_{lt} &= \mathbf{O}_{lt} + \boldsymbol{\epsilon}_{lt}, & (5)\\ \mathbf{O}_{lt} &= \rho\mathbf{O}_{lt-1} + \mathbf{X}_{lt}\boldsymbol{\beta} + \boldsymbol{\eta}_{lt}, & (6) \end{aligned}$$

for all l and t; where, ρ 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 $l = 1, \ldots, r$. 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 ϕ and ν for $\boldsymbol{\eta}_{lt}$.

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

$$\begin{aligned}\log \pi(\boldsymbol{\theta}, \mathbf{z}^*|\mathbf{z}) \quad \propto \quad & -\frac{N}{2}\log\sigma_\epsilon^2 - \frac{1}{2\sigma_\epsilon^2}\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{Z}_{lt} - \mathbf{O}_{lt})'(\mathbf{Z}_{lt} - \mathbf{O}_{lt}) - \frac{N}{2}\log|\sigma_\eta^2 S_\eta| \\ & -\frac{1}{2\sigma_\eta^2}\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{O}_{lt} - \rho\mathbf{O}_{lt-1} - \mathbf{X}_{lt}\boldsymbol{\beta})' S_\eta^{-1}(\mathbf{O}_{lt} - \rho\mathbf{O}_{lt-1} - \mathbf{X}_{lt}\boldsymbol{\beta}) \\ & -\frac{1}{2}\sum_{l=1}^{r}\log|\sigma_l^2 S_0| - \frac{1}{2}\sum_{l=1}^{r}\frac{1}{\sigma_l^2}(\mathbf{O}_{l0} - \boldsymbol{\mu}_l)' S_0^{-1}(\mathbf{O}_{l0} - \boldsymbol{\mu}_l) + \log\pi(\boldsymbol{\theta}). \quad (7)\end{aligned}$$

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 $A = CS_\eta^{-1}$ and C denotes the n by m cross-correlation matrix between the random effects at the n observation locations and m knot locations, $\mathbf{s}_1^*, \ldots, \mathbf{s}_m^*$, and S_η 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 Σ_η is an $m \times m$ matrix which is of much lower dimensional than the same for two previous models GP and AR since we assume that $m << n$.

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 $l = 1, \ldots, r$, where the correlation matrix S_0 is obtained by using the Matérn correlation function in Equation (1) with decay parameter ϕ. Let $\mathbf{w}$ denote the random effects $\mathbf{w}_{lt}$ for $l = 1, \ldots, r$ and $t = 0, 1, \ldots, T$. Let $\boldsymbol{\theta}$ denote all the parameters

$\mathbf{w}$, $\boldsymbol{\beta}$, ρ, σ^2_ϵ, σ^2_w, ϕ, σ^2_l, $l = 1, ..., r$. The logarithm of the joint posterior distribution of the parameters and the missing data is given by:

$$\begin{aligned}\log \pi(\boldsymbol{\theta}, \mathbf{z}^*|\mathbf{z}) \quad \propto \quad & -\frac{N}{2}\log\sigma^2_\epsilon - \frac{1}{2\sigma^2_\epsilon}\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{Z}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta} - A\mathbf{w}_{lt})'(\mathbf{Z}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta} - A\mathbf{w}_{lt}) \\ & -\frac{mrT}{2}\log\sigma^2_\eta - \frac{rT}{2}\log|S_\eta| - \frac{1}{2\sigma^2_\eta}\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{w}_{lt} - \rho\mathbf{w}_{lt-1})'S^{-1}_\eta \\ & (\mathbf{w}_{lt} - \rho\mathbf{w}_{lt-1}) - \frac{m}{2}\sum_{l=1}^{r}\log\sigma^2_l - \frac{r}{2}\log|S_0| - \frac{1}{2}\sum_{l=1}^{r}\frac{1}{\sigma^2_l}\mathbf{w}_{l0}S^{-1}_0\mathbf{w}_{l0} \\ & + \log\pi(\boldsymbol{\theta}). \end{aligned} \tag{10}$$

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., $\boldsymbol{\beta}, \rho$, other than the random effects, are given independent normal prior distributions where the user can specify the means (μ_β, μ_ρ) and variance (δ^2_β, δ^2_ρ). In our illustrations in Section 3 we have taken all those means to be 0 and variance to be 10^{10}, that corresponds to our assumption of flat prior distributions.

The prior distribution for a typical variance parameter is specified through a gamma distribution with mean a/b and variance a/b^2 for the inverse of the variance. 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 ϕ. **spTimer** allows one dimensional grid search for estimating ϕ conditional on fixed values of ν that corresponds to known popular covariance function models such as: exponential for $\nu = 1/2$, and Gaussian when $\nu \to \infty$.

Full Bayesian estimation of ϕ and η is also allowed in **spTimer** corresponding to discrete uniform prior distributions for both these parameters. In all our illustrations the smoothing parameter ν 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 continuous uniform and a Gamma prior distribution with suitable hyper-parameter values for the decay parameter ϕ.

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 decay parameter 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 ϕ and ν 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 ϕ, 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 σ_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 ϕ, i.e., we work with the density of $\log(\phi)$ instead of ϕ since the support of the normal proposal distribution is the real line. In this sampling scheme, keeping ϕ within a range 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,

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

where, $Z_l(\mathbf{s}_i,t)_{rep}$ 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:

$$\pi(Z_l(\mathbf{s}_0,t')|\mathbf{z}) = \int \pi(Z_l(\mathbf{s}_0,t')|O_l(\mathbf{s}_0,t'),\sigma_\epsilon^2)\pi(O_l(\mathbf{s}_0,t')|\boldsymbol{\theta},\mathbf{z}^*)\pi(\boldsymbol{\theta}|\mathbf{z})dO_l(\mathbf{s}_0,t')d\boldsymbol{\theta}d\mathbf{z}^*. \tag{12}$$

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 $\boldsymbol{\theta}^{(j)}$ is drawn from the posterior distribution $\pi(\boldsymbol{\theta},\mathbf{z}^*|\mathbf{z})$ using the details in the model fitting

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 $O_l(\mathbf{s}_1, t), \ldots, O_l(\mathbf{s}_n, t)$. 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 $Z_l^{(j)}(\mathbf{s}_0, t')$, $j = 1, \ldots, J$ where J is a large number, are summarized to make predictive inference. If a transformation, such as the log or the square 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.

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

Table 1: Two main functions in **spTimer**.

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 defines and controls its behavior. For example, `formula` is used to specify the linear part of the model. It is an object of class `formula`, same as that for the `lm` function to fit linear regression models in R. The argument `data` provides the data set to be used for model fitting. 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 segments, for example, T daily observations in each of r years, must be ordered first by year and then by days in each year. The length of the segments must be the same in each year and also for each location. Currently the package cannot handle any irregular time-series data. Missing data should have the standard `NA` identifier. No missing data values are allowed for the covariates.

One required argument `coords` must be supplied as an $n \times 2$ matrix containing the co-ordinates of the n spatial locations, e.g., longitude and latitude, for the model fitting data. One optional argument `knots.coords` is used only under the `GPP` model and, like the `coords` argument, is provided as an $m \times 2$ matrix of coordinates of the m ($< n$) knot locations.

There are also arguments to specify the model to be fitted which can be any one of the three: `GP`, `AR`, and `GPP`. The default model is `GP`.

In `spT.Gibbs` there are also arguments to specify the hyper-parameter values of the prior

distributions. The default value of this argument is `NULL` when automatic values are chosen as flat prior distributions.

The argument `initials` specifies the starting values of 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 as initial values: $\sigma^2_\eta = 0.1$, $\sigma^2_\epsilon = 0.01$. The default value for the spatial decay parameter, ϕ, 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 using estimates from a simple linear model.

It is also possible to specify the method to calculate the distance between any two locations using argument `distance.method`. This can take values `"geodetic:km"`, `"geodetic:mile"` and `"euclidean"` for distance in kilometers, miles and Euclidean distance respectively. The default is `"geodetic:km"`. The argument `cov.fnc` provides the choice of the spatial covariance function and can take any one of the values: `"exponential"`, `"gaussian"`, `"spherical"`, and `"matern"`.

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 annual 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.

There are also arguments to control the running of the MCMC algorithm, for example, `nItr` for the total number of iterations and `nBurn`, the number of burn-in iterations. The full list of arguments and discussion of the output of the function `spT.Gibbs` are given in the manual of **spTimer** (Bakar and Sahu 2013).

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 is for the new coordinate points where we want to predict and `newdata` which defines the covariates. The argument `type` in `predict` is to specify what type of prediction user wants to make by `"spatial"` or `"temporal"`. Detail list of arguments and their descriptions are provided in the **spTimer** manual (Bakar and Sahu 2013).

3.3. Some other useful functions

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

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 `"Gam"` denotes the gamma prior distribution while the abbreviation `"Nor"` denotes the normal prior distribution. For example, the call

```
R> prior <- spT.priors(model="AR", var.prior=Gam(a=2,b=1),
           beta.prior=Nor(0,10^10), rho.prior=Nor(0,10^10),
           phi.prior=Gam(a=2,b=1))
```

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 $\boldsymbol{\beta}$ and the autoregressive parameter ρ will be assigned an independent normal prior distribution with mean 0 and variance 10^{10}. This will also assign a gamma prior distribution for the spatial decay parameter ϕ, see also the documentation for `spatial.decay` where this option can be over written. For fitting other models necessary modifications must be done, more details are provided in the package documentation. A default proper prior with large variance will be assumed if prior for any parameter is missing.

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

The `spT.decay` function is used to specify one of the three possible ways to handle the spatial decay parameter ϕ.

(i) **Fixed**: The first choice is to fix ϕ at a particular value. This is achieved by writing `type="FIXED"` in the argument. For example, for fixing ϕ at 0.01 we write:

```
R> spatial.decay <- spT.decay(type="FIXED", value=0.01)
```

(ii) **Discrete**: This option corresponds to assuming a discrete uniform prior for ϕ in a specified interval. A typical specification is provided below:

```
R> spatial.decay <- spT.decay(type="DISCRETE",
                    limit=c(.01,.02), segments=10)
```

where the `segments` argument specifies the number of support points in the prior distribution under this option. The prior for ϕ in `spT.priors` will be ignored.

(iii) **Metropolis-Hastings**: This specifies the incorporation of a random-walk Metropolis sampling algorithm to sample ϕ. The standard deviation of the normal proposal distribution centered at the current value is to be specified as the `tuning` parameter in a typical call such as:

```
R> spatial.decay <- spT.decay(type="MH", tuning=0.08)
```

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

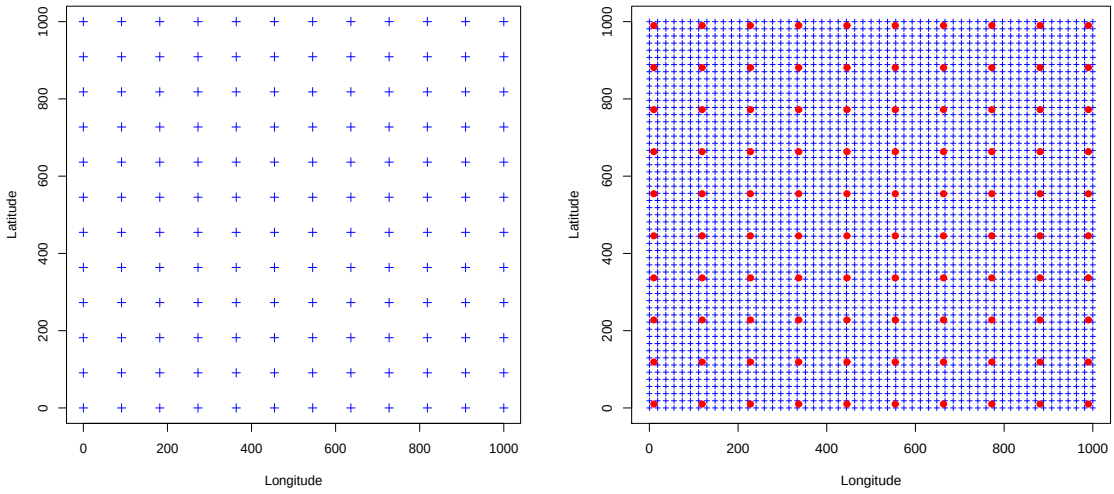
$$\text{VMSE} = \frac{1}{m}\sum_{i=1}^{m}(\widehat{z}_i - z_i)^2, \qquad \text{MAE} = \frac{1}{m}\sum_{i=1}^{m}|\widehat{z}_i - z_i|,$$
$$\text{rBIAS} = \frac{1}{m\bar{z}}\sum_{i=1}^{m}(\widehat{z}_i - z_i), \quad \text{rMSEP} = \sum_{i=1}^{m}(\widehat{z}_i - z_i)^2 / \sum_{i=1}^{m}(\bar{z}_p - z_i)^2,$$

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

4. A simulation example

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. The code for doing this analysis is provided in the accompanying R source file.

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 10×10 square grid inside the range. The temporal domain is taken as 365 days in



(a) (b)

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.

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. See accompanying R code for generating these data.

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 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 the 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 the only covariate effect is captured in the intercept term and we take the true value of this parameter, β_0, to be 5 for all three models. The spatial effect variance, σ^2_η under GP, AR and GPP based models is assumed to be 0.1 that is much larger than the pure error (or nugget effect) variance $\sigma^2_\epsilon = 0.01$. The spatial decay parameter ϕ is assumed to be 0.003 that implies high spatial correlation even at large distances. The temporal auto-correlation parameter, ρ 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 be 100 (see Figure 1(b) for their locations) and we assume initials for $w_l(\mathbf{s}_i, 0)$ as 0 and 0.5 for $l = 1$ under this model.

In this study we illustrate throughout with the Metropolis-Hastings algorithm for simulating 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 2,500 iterations for making inference with first 1,000 iterations as burn-in. As our simulated dataset is moderately large we use the simultaneous fit and predict option in `spT.Gibbs`. Here, to reduce the load in computer's working memory the MCMC output are given in text format (in separate files written to the disk) and then summary results are obtained using R, but by first reading the MCMC samples from the text files back into R. The accompanying file provides the R source code.

4.1. Results

The Gibbs sampler with 2,500 iterations for the GP and AR models took about 13-15 minutes on a personal computer (PC) with core-i5, 2.6GHz processor and 4GB of RAM running a version of the Windows operating system. The running time for the GPP model that fitted more than a million observation was around 35 minutes in the same computer. This shows the viability, within minutes, of the MCMC methods in processing large amounts of space-time data on a standard PC.

The Gibbs sampler produced estimates of the parameters that were very close to the true sim-

ulation values, see Table 2. Moreover, the acceptance rate for sampling the decay parameter ϕ was also reasonable.

The MCMC methods are also able to estimate the missing observations rather well, see Figure 2. This figure plots the actual true values of the observation, pretended to be missing while model fitting using the GPP based model, and their MCMC estimates for the site which had the highest number of missing observations.

We have also considered validation in this example. A typical validation plot of the predictions against observations for the GPP based model is given in Figure 3. Analogous plots for the other models looked similar and are omitted for brevity. We have also experimented with many other modes of validation and calculated the validation statistics defined earlier in Section 3.3, but all those are omitted from this paper. Instead, we provide a small real data example in the next section, taking only a few seconds to implement, which illustrates various aspects of the package **spTimer**.

Model	Parameter	True Value	Mean	95% Interval	Acceptance Rate
GP	β_0	5.0	4.9998	(4.9802, 5.0186)	-
	σ^2_ϵ	0.01	0.0119	(0.0079, 0.0290)	-
	σ^2_η	0.10	0.1493	(0.0533, 0.6060)	-
	ϕ	0.003	0.0046	(0.0012, 0.0088)	35.36%
AR	β_0	5.0	5.1903	(5.1328, 5.2458)	-
	ρ	0.20	0.1696	(0.1607, 0.1783)	-
	σ^2_ϵ	0.01	0.0096	(0.0092, 0.0114)	-
	σ^2_η	0.10	0.1048	(0.0867, 0.2414)	-
	ϕ	0.003	0.0033	(0.0013, 0.0038)	37.76%
GPP	β_0	5.0	5.0754	(5.0086, 5.1571)	-
	ρ	0.20	0.1918	(0.1611, 0.2089)	-
	σ^2_ϵ	0.01	0.0061	(0.0029, 0.0291)	-
	σ^2_η	0.10	0.3318	(0.0788, 1.8962)	-
	ϕ	0.003	0.0022	(0.0001, 0.0036)	32.08%

Table 2: True values of the parameters and their estimates using the summary statistics of the MCMC samples for all three models. The column Mean represents the posterior mean and a 95% credible interval is also provided for each of the parameters. The acceptance rate for sampling the spatial decay parameter ϕ is also given.

5. A practical example

We use the 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 4 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

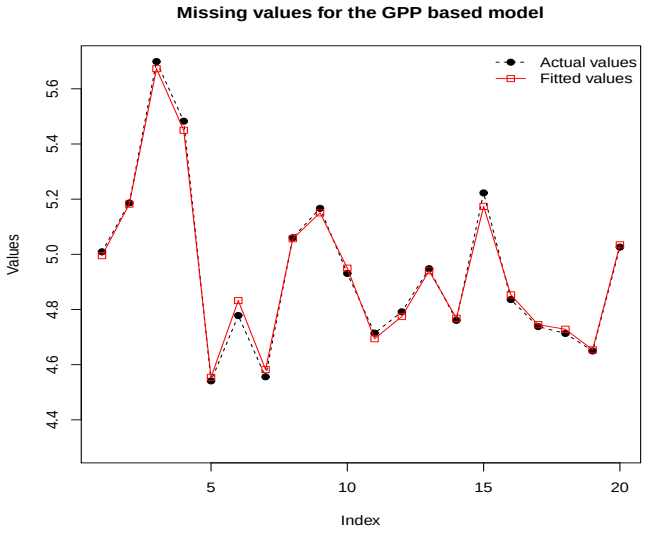


Figure 2: Actual values of the observations, pretended to be missing during model fitting, and their MCMC estimates for the GPP based model for the location that had the highest number of missing observations.

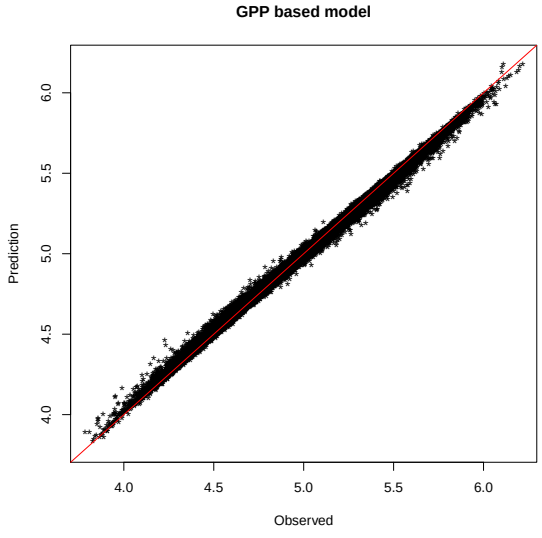


Figure 3: Prediction against observation plots for the simulation example for the GPP based model.

validating the temporal forecasts.

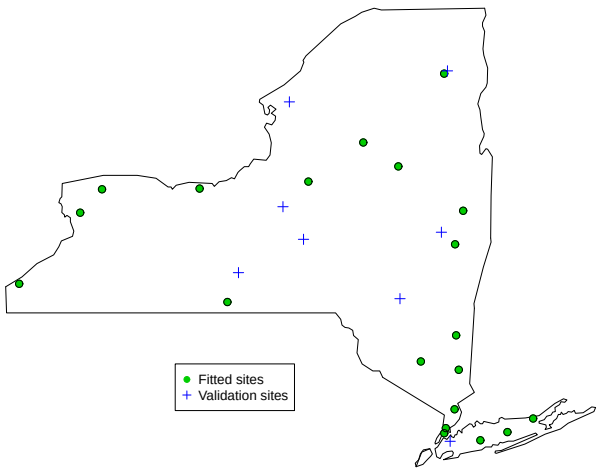


Figure 4: A map of the state of New York showing locations of the 28 ozone monitoring sites of which data from 8 are used for validation purposes.

The data set is distributed with the package **spTimer** where `DataFit` contains the fitting data set from 20 sites and `DataValPred` holds the data from the 8 validation sites. The following set of code lines are used to fit the GP model.

```
R> library("spTimer")
R> data(DataFit);
R> coords<-as.matrix(unique(cbind(DataFit[,2:3])))
R> set.seed(11)
R> post.gp <- spT.Gibbs(formula=o8hrmax ~cMAXTMP+WDSP+RH,
         data=DataFit, coords=coords, model="GP",
         scale.transform="SQRT")
```

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 fomula 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 10,000 further iterations after discarding first 3000. The user manual (Bakar and Sahu 2013) 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: 13000
nBurn: 3000
Acceptance rate for phi (%): 31.51
-----------------------------------------------------
        Goodness.of.fit Penalty   PMCC
values:          252.28  632.69 884.97
-----------------------------------------------------
Computation time: 18.14  - Sec.
```

We can view the MCMC trace plots of the parameters using the familiar `plot` function. One can also obtain residual plots using the function `plot` with the additional argument `residuals=TRUE`. All the MCMC diagnostics of **coda** (Plummer *et al.* 2012) packages are available for analyzing the output of `spT.Gibbs`. For example, one can use the code

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

to generate the autocorrelation plots. However, none of these plots are included here.

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

```
R> summary(post.gp)
Parameters:
               Mean  Median     SD Low2.5p Up97.5p
(Intercept)  3.1341  3.0766 0.8113  1.7131  4.9584
cMAXTMP      0.1352  0.1386 0.0250  0.0758  0.1750
WDSP         0.0772  0.0792 0.0311  0.0096  0.1347
RH          -0.0544 -0.0662 0.0983 -0.2122  0.1773
sig2eps      0.0167  0.0160 0.0042  0.0104  0.0266
sig2eta      0.8243  0.6856 0.4560  0.4799  2.1879
phi          0.0070  0.0073 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 σ^2_η is higher than the nugget effect σ^2_ϵ.

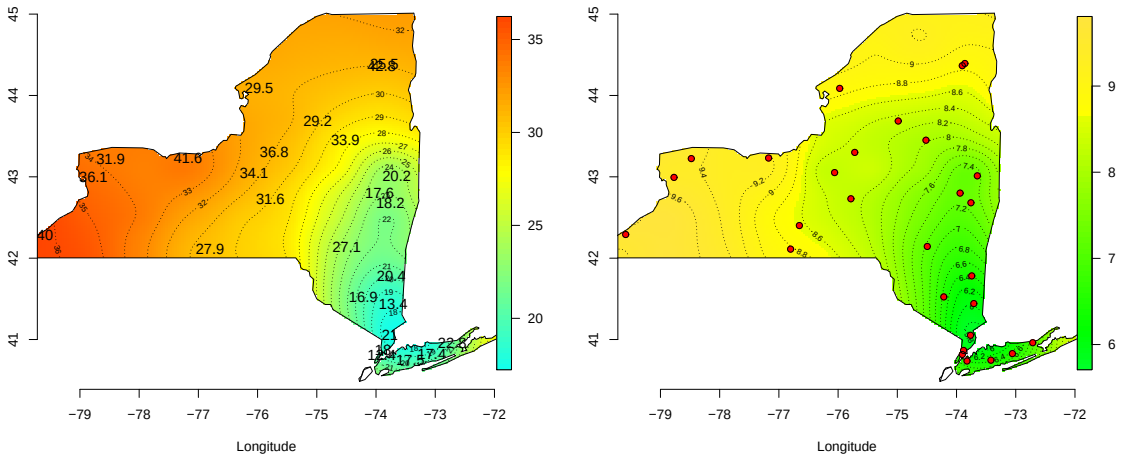
Prediction capabilities of **spTimer** are explored using the `predict` function, as noted in Table 1. After the above model fitting we can use the following code lines to perform and examine prediction.

```
R> data(DataValPred)
R> pred.coords<-as.matrix(unique(cbind(DataValPred[,2:3])))
R> set.seed(11)
R> pred.gp <- predict(post.gp, newdata=DataValPred, newcoords=pred.coords)
```

```
# display description
R> print(pred.gp)
--------------------------------------
Spatial prediction with Model: GP
Covariance function: exponential
Distance method: geodetic:km
Computation time:  8.31  - Sec.
--------------------------------------
R> spT.validation(DataValPred$o8hrmax,c(pred.gp$Median))
   VMSE    RMSE     MAE   rBIAS   rMSEP
42.9560  6.5541  5.0485  0.0136  0.2599
```

where, RMSE is the root mean squared error. The **spTimer** can also perform temporal forecasting at both observed and unobserved locations using the same `predict` function using the additional argument `type="temporal"`, details are provided in the accompanying R source file.

The R source file also contains code for drawing typical predictive maps, see Figure 5 for daily ozone concentration levels and their standard deviations on August 29, 2006.



(a) (b)

Figure 5: 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, that are closely matched by the predictions, and locations are also superimposed in plots (a) and (b) respectively.

We conclude this example with a comparison study with the non-Bayesian generalized additive models (Hastie and Tibshirani 1990) using R package **mgcv** (Wood 2012) as noted by a referee. The GP and AR 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 source file.

For the additive model we use the `gam` function and fit 10 different models and report the predictive performance of the model which performs the best, see Table 3. We observe that the VMSE for the Bayesian space-time GP model is reduced by about 45% compared to the best predictive generalized additive models (GAM), showing superiority of the spatio-temporal models implemented in the package **spTimer**.

Spatial prediction					
Models		VMSE	MAE	rBIAS	rMSEP
Bayesian space-time	GP	42.96	5.05	0.01	0.26
	AR	44.21	5.11	0.01	0.27
non-Bayesian	GAM	77.98	6.88	-0.03	0.46

Table 3: Validation statistics for the GP, AR and the best predictive GAM models.

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. At the present time, 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 written and distributed code has been validated using a substantial simulation example and a real life 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 analyzes 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 co-efficient models. Other possible extensions include the ability to handle data from sensor networks that vary over time and also irregular time series data. 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.

Acknowledgements

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

References

Baddeley A, Turner R (2012). ***spatstat***: *Spatial Point Pattern Analysis, Model-fitting, Simulation, Tests*. R package version 1.30-0, URL http://cran.r-project.org/web/packages/spatstat.

Bakar KS (2012). *Bayesian Analysis of Daily Maximum Ozone Levels*. PhD Thesis, University of Southampton, Southampton, United Kingdom.

Bakar KS, Sahu SK (2013). ***spTimer***: *Spatio-Temporal Bayesian Modelling Using R*. R package version 0.05, URL http://cran.r-project.org/web/packages/spTimer.

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 Royal Statistical Society B*, **70**, 825–848.

Cameletti M, Ignaccolo R, Bande S (2009). "Comparing Air Quality Statistical Models." *Technical Report*.

Campos G, Rodriguez P (2012). ***BLR***: *Bayesian Linear Regression*. R package version 1.3, URL http://cran.r-project.org/web/packages/BLR.

Christensen OF, Ribeiro PJJ (2002). "geoRglm: A Package for Generalised Linear Spatial Models." *R News*, **2**(2), 26–28.

Christensen OF, Ribeiro Jr PJ (2011). ***geoRglm*** - *A Package for Generalised Linear Spatial Models*. R package version 0.9-2, URL http://cran.r-project.org/web/packages/geoRglm.

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.

Diggle P, Ribeiro PJ (2007). *Model-based Geostatistics*. Springer, 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*, **12**(4), 1–24.

Finley AO, Banerjee S, Carlin BP (2013). ***spBayes***: *Univariate and Multivariate Spatial Modeling*. R package version 0.3-1, URL http://cran.r-project.org/web/packages/spBayes.

Furrer R, Nychka D, Sain S (2012). ***fields***: *Tools for Spatial Data*. R package version 6.7, URL http://cran.r-project.org/web/packages/fields.

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 J (2012). ***MCMCglmm**: MCMC Generalised Linear Mixed Models*. R package version 2.17, URL http://cran.r-project.org/web/packages/MCMCglmm.

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 (2011). ***gam**: Generalized Additive Models*. R package version 1.06.2, URL http://cran.r-project.org/web/packages/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.

Liggrs U (2011). ***R2WinBUGS**: Running WinBUGS and OpenBUGS from R / S-PLUS*. R package version 2.1-18, URL http://cran.r-project.org/web/packages/R2WinBUGS.

Martin A, Quinn K, Park J (2012). ***MCMCpack**: Markov chain Monte Carlo (MCMC) Package*. R package version 1.2-4, URL http://cran.r-project.org/web/packages/MCMCpack.

Matérn B (1986). *Spatial Variation*. Second edition. Springer-Verlag, Berlin.

Millo G, Gianfranco P (2012). ***splm**: Econometric Models for Spatial Panel Data*. R package version 1.0-00, URL http://cran.r-project.org/web/packages/splm.

Pebesma E (2013). ***gstat**: Spatial and Spatio-temporal Geostatistical Modelling, Prediction and Simulation*. R package version 1.0-15, URL http://cran.r-project.org/web/packages/gstat.

Pebesma E, Bivand B, Rowlingson B, Rubio VG (2012a). ***sp**: Classes and Methods for Spatial Data*. R package version 1.0-5, URL http://cran.r-project.org/web/packages/sp.

Pebesma E, Graeler B, Gottfried T, Hijmans R (2012b). ***spacetime**: Classes and Methods for Spatial Data*. R package version 1.0-3, URL http://cran.r-project.org/web/packages/spacetime.

Pinheiro J, Bates D, DebRoy S, Sarkar D (2013). ***nlme**: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-107, URL http://cran.r-project.org/web/packages/nlme.

Plummer M (2012). ***rjags**: Bayesian graphical models using MCMC*. R package version 3-9, URL http://cran.r-project.org/web/packages/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/web/packages/coda.

R Development Core Team (2006). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Ribeiro PJJ, Diggle PJ (2001). "**geoR**: Analysis of Geostatistical Data." *R News*, **1**(2), 14–18.

Ribeiro PJJ, Diggle PJ (2012). ***geoR**: Analysis of Geostatistical Data*. R package version 1.7-4, URL http://cran.r-project.org/web/packages/geoR.

Ripley B (2012). ***spatial**: Functions for Kriging and Point Pattern Analysis*. R package version 7.3-5, URL http://cran.r-project.org/web/packages/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, Bakar KS, Awang N (2012). "Bayesian Forecasting Using Hierarchical Spatio-temporal Models with Applications to Ozone Levels in the Eastern United States." Submitted.

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 (1999). *WinBUGS Version 1.2 User Manual*. MRC Biostatistics Unit, 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.

Wood S (2012). ***mgcv:*** *Mixed GAM Computation Vehicle with GCV / AIC / REML smoothness estimation.* R package version 1.7-22, URL http://cran.r-project.org/web/packages/mgcv.

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: $\pi(\boldsymbol{\beta}|.,\mathbf{z}) \sim N(\Delta\chi, \Delta)$, where

$$\Delta^{-1} = \sum_{l=1}^{r}\sum_{t=1}^{T} \mathbf{X}'_{lt}\Sigma_\eta^{-1}\mathbf{X}_{lt} + \mathbf{I}_p/\delta_\beta^2$$

$$\chi = \sum_{l=1}^{r}\sum_{t=1}^{T} \mathbf{X}'_{lt}\Sigma_\eta^{-1}\mathbf{O}_{lt}$$

- Similarly from (4), we sample σ_ϵ^2 and σ_η^2 from the following conditional distributions respectively:

$$\pi(1/\sigma_\epsilon^2|.,\mathbf{z}) \sim G\left(\frac{N}{2} + a, b\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{Z}_{lt} - \mathbf{O}_{lt})'(\mathbf{Z}_{lt} - \mathbf{O}_{lt})\right),$$

$$\pi(1/\sigma_\eta^2|.,\mathbf{z}) \sim G\left(\frac{N}{2} + a, b\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta})'\Sigma_\eta^{-1}(\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta})\right)$$

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

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

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

- We can not find any closed form for the posterior distribution of ϕ. Hence, we sample ϕ from the following kernel:

$$\pi(\phi|.,\mathbf{z}) \propto \pi(\phi) \times |\mathbf{S}_\eta|^{-rT/2} \times \exp\left[\frac{1}{2}\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta})'\Sigma_\eta^{-1}(\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta})\right]$$

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: $\pi(\boldsymbol{\beta}|.,\mathbf{z}) \sim N(\Delta\chi, \Delta)$, where:

$$\Delta^{-1} = \sum_{l=1}^{r}\sum_{t=1}^{T} \mathbf{X}'_{lt}\Sigma_{\eta}^{-1}\mathbf{X}_{lt} + \mathbf{I}_p/\delta_{\beta}^2$$

$$\chi = \sum_{l=1}^{r}\sum_{t=1}^{T} \mathbf{X}'_{lt}\Sigma_{\eta}^{-1}(\mathbf{O}_{lt} - \rho\mathbf{O}_{lt})$$

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

$$\Delta^{-1} = \sum_{l=1}^{r}\sum_{t=1}^{T} \mathbf{O}'_{lt}\Sigma_{\eta}^{-1}\mathbf{O}_{lt} + \mathbf{I}_p/\delta_{\rho}^2$$

$$\chi = \sum_{l=1}^{r}\sum_{t=1}^{T} \mathbf{O}'_{lt}\Sigma_{\eta}^{-1}(\mathbf{O}_{lt} - \mathbf{X}_{lt}\boldsymbol{\beta})$$

- For σ_{ϵ}^2 and σ_{η}^2 we sample from the following conditional distributions respectively:

$$\pi(1/\sigma_{\epsilon}^2|.,\mathbf{z}) \sim G\left(\frac{N}{2} + a, b\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{Z}_{lt} - \mathbf{O}_{lt})'(\mathbf{Z}_{lt} - \mathbf{O}_{lt})\right),$$

$$\pi(1/\sigma_{\eta}^2|.,\mathbf{z}) \sim G\left(\frac{N}{2} + a, b\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{O}_{lt} - \rho\mathbf{O}_{lt-1} - \mathbf{X}_{lt}\boldsymbol{\beta})'\Sigma_{\eta}^{-1}(\mathbf{O}_{lt} - \rho\mathbf{O}_{lt-1} - \mathbf{X}_{lt}\boldsymbol{\beta})\right)$$

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

  Case 1:

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

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

  Case 2:

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

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

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

$$\begin{aligned}\pi(\phi|.,\mathbf{z}) \quad \propto \quad & \pi(\phi) \times |\mathbf{S}_{\eta}|^{-rT/2} \times \\ & \exp\left[\frac{1}{2}\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{O}_{lt} - \rho\mathbf{O}_{lt} - \mathbf{X}_{lt-1}\boldsymbol{\beta})'\Sigma_{\eta}^{-1}(\mathbf{O}_{lt} - \rho\mathbf{O}_{lt-1} - \mathbf{X}_{lt}\boldsymbol{\beta})\right] \\ & \times|\mathbf{S}_0|^{-r/2} \times \exp\left[\frac{1}{2}\sum_{l=1}^{r}\frac{1}{\sigma_l^2}(\mathbf{O}_{l0} - \boldsymbol{\mu}_l)'S_0^{-1}(\mathbf{O}_{l0} - \boldsymbol{\mu}_l)\right]\end{aligned}$$

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,

$$\Delta^{-1} = \frac{1}{\sigma_\epsilon^2}\sum_{l=1}^{r}\sum_{t=1}^{T} X_{lt}'X_{lt} + \mathbf{I}_p/\delta_\beta^2,$$

$$\chi = \frac{1}{\sigma_\epsilon^2}\sum_{l=1}^{r}\sum_{t=1}^{T} X_{lt}'(\mathbf{Z}_{lt} - A\mathbf{w}_{lt}).$$

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

$$\Delta^{-1} = \sum_{l=1}^{r}\sum_{t=1}^{T} \mathbf{w}_{lt-1}'\Sigma_\eta^{-1}\mathbf{w}_{lt-1} + \mathbf{I}_p/\delta_\rho^2$$

$$\chi = \sum_{l=1}^{r}\sum_{t=1}^{T} \mathbf{w}_{lt-1}'\Sigma_\eta^{-1}\mathbf{w}_{lt}.$$

- The variance parameters σ_ϵ^2 and σ_η^2 are sampled from:

$$G\left(\frac{N}{2} + a, b + \frac{1}{2}\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{Z}_{lt} - X_{lt}\boldsymbol{\beta} - A\mathbf{w}_{lt})'(\mathbf{Z}_{lt} - X_{lt}\boldsymbol{\beta} - A\mathbf{w}_{lt})\right)$$

$$G\left(\frac{mrT}{2} + a, b + \frac{1}{2}\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{w}_{lt} - \rho\mathbf{w}_{lt-1})'\Sigma_\eta^{-1}(\mathbf{w}_{lt} - \rho\mathbf{w}_{lt-1})\right)$$

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

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

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

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

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

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

- The full conditional distribution of ϕ is not available in closed form. The log of the conditional posterior density (upto an additive constant) is given by:

$$\log\pi(\phi|...) \quad \propto \quad \pi(\phi) \times |S_\eta|^{rT} \times \exp\left[\frac{1}{2}\sum_{l=1}^{r}\sum_{t=1}^{T}(\mathbf{w}_{lt} - \rho\mathbf{w}_{lt-1})'\Sigma_\eta^{-1}(\mathbf{w}_{lt} - \rho\mathbf{w}_{lt-1})\right]$$
$$\times|S_0|^{r/2} \times \exp\left[\frac{1}{2}\sum_{l=1}^{r}\frac{1}{\sigma_l^2}\mathbf{w}_{l0}\mathbf{S}_0^{-1}\mathbf{w}_{l0}\right].$$

Affiliation:

Khandoker Shuvo Bakar
CSIRO Mathematics, Informatics and Statistics
GPO Box 664, Canberra, ACT 2601, Australia.
E-mail: shuvo.bakar@csiro.au
URL: http://www.cmis.csiro.au/

Sujit Kumar Sahu
Southampton Statistical Sciences Research Institute
University of Southampton
Southampton, SO17 1BJ, UK.
E-mail: S.K.Sahu@soton.ac.uk
URL: http://www.soton.ac.uk/~sks/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd