Bayesian Nonstationary Gaussian Process Models for Large Datasets via Treed Process Convolutions

Waley W. J. Liang and Herbert K. H. Lee
University of California, Santa Cruz

Abstract

Spatial modeling often relies upon stationary Gaussian processes (GPs), but the assumption that the correlation structure is independent of the spatial location is invalid in many applications. Various nonstationary GP models have been developed to solve this problem, however, many of them become impractical when the sample size is large. To tackle this problem, we develop a process convolutions-based GP model by convolving a smoothing kernel with a partitioned latent process. Nonstationarity in the GP is obtained by allowing the variability of the latent process and the kernel size to change across partitions. Partitioning is achieved using a method similar to that of Classification and Regression Trees, which results in a binary tree structure. A Bayesian approach is used to simultaneously guide the partitioning process and estimate the parameters of the treed model.

Keywords: Spatial Statistics; Classification and Regression Trees; Markov chain Monte Carlo

1 Introduction

Gaussian spatial models are widely adopted in many applications such as geology, climatology, hydrology, and computer simulation experiments. The phenomenon of interest is described using a Gaussian process (GP). It is a convenient method for modeling data indexed by spatial location (e.g., point-referenced data such as observed precipitation count referenced by the location of measurement over a bounded domain) and can capture much of the spatial behavior based on the specification of the correlation structure. In many cases, the mean value and correlation structure are assumed to be homogeneous across locations, that is, the mean value and the correlation between any two points in the GP do not depend on locations. The mathematical term for this behavior is *stationarity.* The stationary assumption may be valid in some applications, but proves to be untenable in many cases. In fact, most spatial phenomena exhibit a correlation structure that depends on the locations. Another issue in spatial statistics is that recent advances in geographical information systems and global positioning systems enable the formation of large spatial datasets. However, inversion of the covariance matrix makes standard Gaussian process approaches infeasible for large dataseets. Developing efficient nonstationary spatial models for large datasets has been one of the core interests of spatial statisticians. We introduce herein a new approach that combines process convolutions with partitioning processes, creating a new model which is nonstationary and computationally efficient enough to use for large datasets.

1.1 Literature review

There are several categories of approaches for modeling nonstationarity in moderate-to-large spatial datasets. One popular category approximates the underlying process of interest using process convolutions, low rank splines or basis functions (Higdon, 2002; Paciorek and Schervish, 2006; Wikle and Cressie, 1999; Lin et al., 2000; Hoef et al., 2004; Xia and Gelfand, 2006; Kammann and Wand, 2003; Paciorek, 2007). The key idea is to reduce the dimension of the problem, making the computational complexity depend on the number of bases instead of the sample size, where the former is often significantly smaller than the latter. For the process convolutions approach, the process of interest is modeled by convolving a smoothing kernel with a latent process. Nonstationarity can be induced by either varying the kernel spatially while fixing the latent process (Higdon et al., 1999; Lemos and Sansó, 2009), or by varying the dependence structure of the latent process while fixing the kernel (Fuentes and Smith, 2001; Lee et al., 2005). The drawback of varying the kernels over space is that some parameters of the kernel re-parametrization have to be fixed and overfitting and mixing problems are found when these parameters are allowed to vary (Swall, 1999). The approach by Fuentes and Smith (2001) requires defining a set of local regions wherein a stationary process is assumed for each; then this set of stationary processes are treated as the latent process in the convolution. Many of these approaches become computationally difficult on large datasets; others require large amounts of user specification. One recently developed model that is similar to the process convolutions approach is the predictive process model by Banerjee et al. (2008). It defines a discrete process by fixing a grid of knots over the spatial domain and assuming the values (dependent variable) at those knots to follow a Multivariate Gaussian distribution with a specific correlation function. Then, the process of interest is modeled with a GP whose covariance is a transformation of the covariance of the discrete process. Although this approach is nonstationary by construction, it is limited to situations where the process of interest is nonstationary overall but locally stationary with region-specific anisotropy. In this case, the nonstationary Matérn correlation can be used by defining separate spatially varying parameters for each region. However, this requires knowing the specific regions in advance and such information may not be easily available in many applications.

A second approach uses partitioning to achieve nonstationarity, fitting independent GPs within each region. (Kim et al., 2005; Gramacy and Lee, 2008). Inference of the model parameters is carried out using Markov chain Monte Carlo (MCMC). The drawback of this approach is that each MCMC iteration still requires a matrix decomposition whose complexity increases at a rate of cube of the sample size, which renders this modeling approach impractical for large datasets.

The third approach is nonparametric methods. This includes Multivariate Adaptive Regression Splines (MARS) (Friedman, 1991), which models the data using the sum of a set of basis functions (hinge functions) with different weights. Although the computational speed is fast, fitting MARS to nonstationary data usually results in large residual errors. Another notable model is the deformation approach of (Sampson and Guttorp, 1992). The observations in the original domain are viewed as a nonlinear transformation of points in virtual domain wherein stationarirty can be asssumed. Bayesian versions have been pursued by (Damian et al., 2001) where the transformation is implemented using thin plate splines, and by (Schmidt and O'Hagan, 2003) where the transformation is explained using a bivariate GP. Nonstationarity is introduced through a nonparametric specification of the covariance function. Recently, Gelfand et al. (2005) have proposed a spatial Dirichlet process (SDP) mixture model to produce a random spatial process that is neither Gaussian nor stationary. Essentially, it is a Dirichlet process defined on a space of surfaces and adopts the distribution of a stochastic process as its base measure. The realizations are discrete probability measures with countable support with probability one. Although in principle it can capture virtually any distribution for the observables, the way inference has to be done is not that satisfactory because it insists that given the countable collection of surfaces, only one realization is taken as the model surface. Improvement has been introduced by Duan et al. (2005), where a random distribution is placed on the spatial effects that allows different surface selection at different sites. However, this improvement renders the model computationally more demanding and may not be suitable for large datasets.

1.2 Motivation

Among the models mentioned above, the kernel convolutions/basis functions approach is generally more efficient than the other two categories in terms of computationally speed. Although the partitioning GP models can alleviate this disadvantage to a certain degree (since the inversion of the sub-covariance matrix in each partition is faster than the inversion of the joint covariance matrix), the computational drawback is inevitable as the sample size becomes massive. On the other hand, process convolutions models require estimation of a set of latent variables referenced over the spatial domain, and the number of such variables increases as the dimension of the domain grows. This is the primary drawback of process convolutions. However, many applications, such as geological and environmental applications, have a domain dimension of only two (longitude and latitude) or three with a large dataset. In such case, the standard GP approach is simply impractical, and process convolutions is a more suitable choice. In this paper, we present a nonstationary GP model based on the process convolutions approach. The latent process in the convolution is partitioned using a tree generating procedure similar to that of Bayesian CART (Classification and Regression Trees) Models (Chipman et al., 1998). Nonstationarity in the GP is induced by two ingredients. First, allowing the variability of the latent process to change across partitions induces nonstationarity at the global level since every response depends on the full set of latent process points. This makes the correlation between any two points in the GP to depend on their respective locations. Second, allowing the kernel shape and size to change across partitions captures local patterns that can not be explained at the global level by varying the latent process. Under this setup, a Bayesian approach and specifically Reversible Jump Markov chain Monte Carlo (RJ-MCMC) (Green, 1995) is used to explore the treed model space and estimate the respective parameters simultaneously.

2 Process Convolutions based GP models

2.1 Gaussian process

A common random field used in spatial modeling is the Gaussian random field, often referred to as a Gaussian process, which is a collection of random variables $\{z(\mathbf{s}) : \mathbf{s} \in \mathbb{R}^d\}$ such that for any finite set of locations $\{\mathbf{s}_1, \cdots, \mathbf{s}_n : \mathbf{s}_i \in \mathbb{R}^d\}$, the joint distribution of $\mathbf{z} = \{z(\mathbf{s}_1), \cdots, z(\mathbf{s}_n)\}$ follows a multivariate Gaussian distribution. A GP can be completely specified by its *mean function* $\mu(\mathbf{s}) = E[z(\mathbf{s})]$, and its *covariance function*, $C(\mathbf{s}_i, \mathbf{s}_{i'}) = Cov(z(\mathbf{s}_i), z(\mathbf{s}_{i'}))$. Generally, the dependence structure and smoothness in z are governed by the choice of the covariance function. A common simplifying assumption of GP models is stationarity. The strictest form of stationarity requires that the distribution of any finite collection of points in the process to be invariant under translation. A weaker version, *second-order stationarity*, is usually a more pragmatic and sufficient assumption. It requires the existence of the mean and marginal variance to be independent of location, and the covariance between any two points, $z(\mathbf{s}_i)$ and $z(\mathbf{s}_{i'})$, to depend only on the vector difference of their locations $\mathbf{s}_i - \mathbf{s}_{i'}$. Isotropy, an even more simplifying assumption, further requires that the covariance between any two points depends only on their distance $||\mathbf{s}_i - \mathbf{s}_{i'}||$. Common isotropic correlation functions include the Gaussian, Exponential, Matérn, and the Spherical classes (Cressie, 1991; Stein, 1999; Banerjee et al., 2003).

2.2 Process Convolutions

An alternative specification of a stationary (second-order) Gaussian process can be done via process convolutions (Higdon, 1998, 2002) as follows,

$$z(\mathbf{s}) = \int_{\mathcal{S}} k(\mathbf{u} - \mathbf{s}) x(\mathbf{u}) d\mathbf{u}, \quad \mathbf{s}, \mathbf{u} \in \mathcal{S} \subseteq \mathbb{R}^d,$$

where $x(\cdot)$ is a White noise process with mean zero and precision λ, and $k(\cdot)$ is a symmetric kernel (e.g, Gaussian) defined over $\mathcal{S} \subseteq \mathbb{R}^d$. The expectation of z is 0 and the covariance is given by

$$Cov(z(\mathbf{s}_i), z(\mathbf{s}_{i'})) \quad = \quad \lambda^{-1} \int_{\mathcal{S}} k(\mathbf{u} - \mathbf{h}) k(\mathbf{u}) d\mathbf{u}, \quad \mathbf{h} = \mathbf{s}_i - \mathbf{s}_{i'}.$$

That is, the covariance depends only on the vector difference $\mathbf{h} = \mathbf{s}_i - \mathbf{s}_{i'}$, and is actually the result of convolving the kernel with itself. In practice, we would approximate the above convolution using a finite set of regularly spaced basis points $\mathbf{u}_1, \cdots, \mathbf{u}_m \in \mathcal{S}$ with $x(\mathbf{u}_j) \sim N(0, \lambda^{-1})$. A discrete approximation of $z(\mathbf{s})$ can be obtained as $z(\mathbf{s}) \approx \sum_{j=1}^{m} k(\mathbf{u}_j - \mathbf{s})x(\mathbf{u}_j)$. In most applications, only a finite set of spatial sites $\mathbf{s}_1, \cdots, \mathbf{s}_n \in \mathcal{S}$ are of interest. In such cases, the above representation can be written in matrix form as $\mathbf{z} = \mathbf{K}\mathbf{x}$, where $\mathbf{z} = (z(\mathbf{s}_1), \cdots, z(\mathbf{s}_n))^\top$, $\mathbf{x} = (x(\mathbf{u}_1), \cdots, x(\mathbf{u}_m))^\top$, and $\mathbf{K}$ is a $(n \times m)$ matrix with elements $\mathbf{K}_{ij} = k(\mathbf{u}_j - \mathbf{s}_i)$.

3 Treed Models

In general, treed models use binary trees to partition the predictor space into disjoint subsets in which the distribution of the response variable becomes more homogeneous and a simpler submodel can be fitted in each partition. Classification and Regression Trees (CART) by Breiman et al. (1984) is a popular example of a treed model. Bayesian formulations of CART can be found in Chipman et al. (1998) and Denison et al. (1998), where the former presents a tree generating process which we also use in our model. This tree generating process is done in a binary and recursive manner, which results in a tree structure such that each internal node is associated with a splitting rule that determines the location of partitioning. Each measurement of the response variable is assigned to only one of the terminal nodes (partitions) and is assumed to follow the sampling distribution associated with that node. Recursively splitting results in a binary tree $\mathcal{T}$ with b terminal nodes and each terminal node is associated with a subset, denoted by $\{\mathbf{F}_\nu, \mathbf{y}_\nu\}$, of the full data set $\{\mathbf{F}, \mathbf{y}\}$ such that $\mathbf{F} = (\mathbf{F}_1^\top, \cdots, \mathbf{F}_b^\top)^\top$ and $\mathbf{y} = (\mathbf{y}_1^\top, \cdots, \mathbf{y}_b^\top)^\top$. The tree $\mathcal{T}$ associates a separate model for each partition, that is, the sampling distribution of $\mathbf{y}_\nu$ is often described by a separate parametric model $\mathrm{p}(\mathbf{y}_\nu|\mathbf{F}_\nu, \boldsymbol{\theta}_\nu, \mathcal{T})$, where $\boldsymbol{\theta}_\nu$ denotes the parameter vector associated with partition ν. Letting $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_b)$, a treed model is fully characterized by $(\boldsymbol{\Theta}, \mathcal{T})$. An important assumption of treed models is that data within the same partition are i.i.d. and data across partitions are independent. Hence, the sampling distribution of $\mathbf{y}$ can obtained as

$$\mathrm{p}(\mathbf{y}|\mathbf{F}, \boldsymbol{\Theta}, \mathcal{T}) = \prod_{\nu=1}^{b} \mathrm{p}(\mathbf{y}_\nu|\mathbf{F}_\nu, \boldsymbol{\theta}_\nu, \mathcal{T}). \tag{1}$$

Implementation of treed models via a Bayesian approach requires prior specification for $(\mathbf{\Theta}, \mathcal{T})$. Because $\mathbf{\Theta}$ depends on the tree structure (e.g., number of terminal nodes), it would be convenient to use the relationship $\mathrm{p}(\mathbf{\Theta}, \mathcal{T}) = \mathrm{p}(\mathbf{\Theta}|\mathcal{T})\mathrm{p}(\mathcal{T})$ and specify the tree prior $\mathrm{p}(\mathcal{T})$ and the conditional parameter prior $\mathrm{p}(\mathbf{\Theta}|\mathcal{T})$ separately. Following Chipman et al. (1998), $\mathrm{p}(\mathcal{T})$ is specified through an implicit tree generating process so that each realization from this process is considered a random draw from this prior; $\mathrm{p}(\mathbf{\Theta}|\mathcal{T})$ is specified by imposing conjugate priors on terminal node parameters while assuming conditional independence of parameters across terminal nodes. More details of prior specification will be discussed in the formulation of our treed process convolutions GP model in the next section. Recent work making use of Bayesian treed models includes fitting a linear model (Chipman et al., 2002), or a Gaussian process model (Gramacy and Lee, 2008) in each partition.

4 Treed Process Convolutions GP model

Here we use a measurement error formulation, rather than the nugget formulation of geostatistics. Nonstationarity in the model is induced by partitioning the latent process and assuming a separate correlation structure for each partition. Moreover, the kernel precision matrices are allowed to vary across partitions so that local patterns can be better captured. The partitioning methodology follows that of Bayesian CART. A Bayesian approach is used to explore the model space and estimate the respective parameters.

4.1 Model setup

Denote the observation at location $\mathbf{s}$ by $y(\mathbf{s})$, where $\mathbf{s} \in \mathcal{S} \subseteq \mathbb{R}^d$. In practice, $y(\mathbf{s})$ is often decomposed as

$$y(\mathbf{s}) = \mu(\mathbf{s}) + z(\mathbf{s}) + \epsilon(\mathbf{s}), \tag{2}$$

where $\mu(\mathbf{s})$ denotes the mean function, $z(\mathbf{s})$ denotes the value of the underlying zero-mean stochastic process at location $\mathbf{s}$, and $\epsilon(\mathbf{s}) \sim N(0, \phi^{-1})$ denotes the Gaussian measurement error at $\mathbf{s}$ with ϕ being the precision. In general, the mean function is specified as a constant or a linear function of the covariates (spatial locations and/or attributes). Although the mean function can be specified in higher order, complicated trend characteristics are usually better described through the stochastic component z, which is represented by process convolutions: $z(\mathbf{s}) = \int_{\mathcal{S}} k(\mathbf{u}-\mathbf{s})x(\mathbf{u})d\mathbf{u}$. In the standard Bayesian GP model setting, the stochastic component z is usually given a zero-mean GP prior. In the process convolutions approach, this is equivalent to imposing a White noise process prior on x. The covariance between $z(\mathbf{s}_i)$ and $z(\mathbf{s}_{i'})$ is given by

$$Cov(z(\mathbf{s}_i), z(\mathbf{s}_{i'})) = \int_{\mathcal{S}}\int_{\mathcal{S}} k(\mathbf{s}_i - \mathbf{u}_j)k(\mathbf{s}_{i'} - \mathbf{u}_{j'})Cov(x(\mathbf{u}_j), x(\mathbf{u}_{j'}))d\mathbf{u}_j d\mathbf{u}_{j'}. \tag{3}$$

Suppose that $\mathcal{S}$ is partitioned into b disjoint regions $\{\mathcal{S}_\nu : \nu = 1, \cdots, b\}$, and assuming that each partition of x has a separate covariance structure. Then, z is clearly nonstationary except when $Cov(x(\mathbf{u}_j), x(\mathbf{u}_{j'})) = 0$ for $j \neq j'$ and $Cov(x(\mathbf{u}_j), x(\mathbf{u}_{j'})) = c$ for all $j = j'$, where c is a constant. This extreme case happens only when x is a White noise process. This shows that partitioning the latent process x creates a more flexible GP model such that nonstationarity (or not) of the resulting process z depends on the covariance structures of the latent process x, which is governed by partitioning.

In practice, we use the discrete version of process convolutions. Fixing a finite set of regularly spaced background points $\mathbf{u}_1, \cdots, \mathbf{u}_m \in \mathcal{S}$, z can be approximated by $z(\mathbf{s}) = \sum_{j=1}^{m} k(\mathbf{u}_j - \mathbf{s})x(\mathbf{u}_j)$. As mentioned before, discrete process convolutions is nonstationary by construction. Partitioning generalizes this model by allowing more flexibility to the background points x. Given a finite set of samples $\{y(\mathbf{s}_i) : \mathbf{s}_i \in \mathcal{S}, i = 1, \cdots, n\}$, the model can be written using vector notation,

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{z} + \boldsymbol{\epsilon}, \tag{4}$$

where $\boldsymbol{\mu} = \mathbf{F}\boldsymbol{\beta}$ such that $\mathbf{F}$ is a $(n \times (p+1))$ design matrix whose first column contains all 1's representing the intercept, and $\boldsymbol{\beta}$ is a $((p+1)\times 1)$ linear coefficient vector; the vector $\mathbf{z}$ and $\boldsymbol{\epsilon}$ contain the values of the process z and ϵ, respectively, at locations $\{\mathbf{s}_1, \cdots, \mathbf{s}_n\}$, i.e., $\mathbf{z} = (z(\mathbf{s}_1), \cdots, z(\mathbf{s}_n))^\top$, and $\boldsymbol{\epsilon} = (\epsilon(\mathbf{s}_1), \cdots, \epsilon(\mathbf{s}_n))^\top$. As shown in Section 2.2, $\mathbf{z}$ can be written in matrix form as $\mathbf{z} = \mathbf{K}\mathbf{x}$, where $\mathbf{x} = (x(\mathbf{u}_1), \cdots, x(\mathbf{u}_m))^\top$ and $\mathbf{K}$ is a $(n \times m)$ kernel matrix with elements $\mathbf{K}_{ij} = k(\mathbf{u}_j - \mathbf{s}_i)$. Suppose that $\mathcal{S}$ is partitioned into b disjoint regions $\{\mathcal{S}_\nu : \nu = 1, \cdots, b\}$, we assume a separate mean function $\boldsymbol{\mu}_\nu$, latent process component $\mathbf{x}_\nu$, observation error precision ϕ_ν, and kernel precision matrix $\mathbf{Q}_\nu$ for each partition. Specifically, each basis point $\mathbf{u}$ is treated as the center of a kernel, and for all $\mathbf{u}$'s in a partition, the corresponding kernels have the same precision matrix. This leads to the partitioning of $\{\mathbf{y}, \boldsymbol{\mu}, \mathbf{F}, \mathbf{z}, \mathbf{K}, \mathbf{x}, \boldsymbol{\epsilon}\}$ such that

$$\begin{aligned}
&\mathbf{y} = (\mathbf{y}_1^\top, \cdots, \mathbf{y}_b^\top)^\top, \quad \boldsymbol{\epsilon} = (\boldsymbol{\epsilon}_1^\top, \cdots, \boldsymbol{\epsilon}_b^\top)^\top, \quad \boldsymbol{\epsilon}_\nu \sim N(0, \phi_\nu^{-1}\mathbf{I}_{n_\nu}),\\
&\boldsymbol{\mu} = (\boldsymbol{\mu}_1^\top, \cdots, \boldsymbol{\mu}_b^\top)^\top, \quad \mathbf{F} = (\mathbf{F}_1^\top, \cdots, \mathbf{F}_b^\top)^\top, \quad \boldsymbol{\mu}_\nu = \mathbf{F}_\nu \boldsymbol{\beta}_\nu\\
&\mathbf{z} = (\mathbf{z}_1^\top, \cdots, \mathbf{z}_b^\top)^\top, \quad \mathbf{K} = (\mathbf{K}_1^\top, \cdots, \mathbf{K}_b^\top)^\top, \quad \mathbf{x} = (\mathbf{x}_1^\top, \cdots, \mathbf{x}_b^\top)^\top, \quad \mathbf{z}_\nu = \mathbf{K}_\nu \mathbf{x}_\nu
\end{aligned}$$

where n_ν denotes the number of observations in $\mathcal{S}_\nu$, $\mathbf{I}_{n_\nu}$denotes the $(n_\nu \times n_\nu)$ identity matrix, and $\{\mathbf{y}_\nu, \boldsymbol{\mu}_\nu, \mathbf{F}_\nu, \boldsymbol{\beta}_\nu, \mathbf{z}_\nu, \mathbf{K}_\nu, \mathbf{x}_\nu, \boldsymbol{\epsilon}_\nu, \phi_\nu\}$ are the corresponding matrix/vector components associated with $\mathcal{S}_\nu$. Partitions are generated by recursively choosing a dimension within the parent partition and split at one of the available values. Splitting is allowed only within the parent partition, i.e., the split can not go beyond the boundary of the parent partition. This results in a binary tree structure as discussed in Section 3 where the internal nodes represent the parent partitions generated during the splitting process and the terminal nodes represent the final partitions. Together, a chosen dimension and the respective splitting value forms a splitting rule, and each internal node of the treed model is associated with a unique splitting rule. We denote the set of all splitting rules in the treed model by $\boldsymbol{\rho}$ and the treed model itself by $\mathcal{T}$. Under this construction, the sampling distribution of $\mathbf{y}_\nu$ is given by

$$\mathbf{y}_\nu | \mathbf{x}, \boldsymbol{\beta}_\nu, \phi_\nu, \boldsymbol{\rho}, \mathcal{T}, \mathbf{K}_\nu, \mathbf{F}_\nu, \quad \sim \quad N_{n_\nu}(\mathbf{F}_\nu \boldsymbol{\beta}_\nu + \mathbf{K}_\nu \mathbf{x}\,, \phi_\nu^{-1}\mathbf{I}_{n_\nu}). \tag{5}$$

Following the convention used in Classification and Regression Trees, terminal node parameters are assumed to be independent conditional on the tree structure. The full likelihood of the treed model $\mathcal{T}$ and its parameters is given by

$$\begin{aligned}
L(\mathbf{x}, \boldsymbol{\beta}, \boldsymbol{\phi}, \boldsymbol{\rho}, \mathcal{T} | \mathbf{y}, \mathbf{K}, \mathbf{F}) &= \prod_{\nu=1}^{b} N_{n_\nu}(\mathbf{F}_\nu \boldsymbol{\beta}_\nu + \mathbf{K}_\nu \mathbf{x}\,, \phi_\nu^{-1}\mathbf{I}_{n_\nu})\\
&= \prod_{\nu=1}^{b} \Big(\frac{\phi_\nu}{2\pi}\Big)^{n_\nu/2} \exp\Big\{ -\frac{\phi_\nu}{2}(\mathbf{y}_\nu - \mathbf{z}_\nu)^\top (\mathbf{y}_\nu - \mathbf{z}_\nu)\Big\},
\end{aligned} \tag{6}$$

where $\mathbf{z}_\nu = \mathbf{F}_\nu \boldsymbol{\beta}_\nu + \mathbf{K}_\nu \mathbf{x}$ and $\boldsymbol{\phi} = \{\phi_1, \cdots, \phi_b\}$.

4.2 Bayesian estimation

Given a treed model $\mathcal{T}$ with b partitions, denote the set of unknown parameters by $\boldsymbol{\Theta} = \{\{\mathbf{x}_\nu, \boldsymbol{\beta}_\nu, \phi_\nu\}_{\nu=1}^b, \boldsymbol{\rho}\}$. We follow a Bayesian approach to explore the posterior space of treed models and make inference about the model parameters. This requires a prior specification for $(\boldsymbol{\Theta}, \mathcal{T})$. Following Chipman et al. (1998), we use the conditional relationship $P(\boldsymbol{\Theta}, \mathcal{T}) = P(\boldsymbol{\Theta}|\mathcal{T})P(\mathcal{T})$ to specify the priors separately. First, $P(\mathcal{T})$ is specified through a tree generating process as follows

1. Initialize $\mathcal{T}$ consisting of a single root node denoted η.
2. Split the terminal node η with probability $P_{split}(\eta, \mathcal{T})$.
3. Assign a splitting rule ρ with probability $P_{rule}(\rho|\eta, \mathcal{T})$ to η if it splits, and create its left and right children nodes (new terminal nodes). Let $\mathcal{T}$ denote this new tree and repeat steps 2 and 3.

The probability of splitting a node η is given by $P_{split}(\eta, \mathcal{T}) = \alpha_T(1 + d_\eta)^{-\beta_T}$, where d_η is the number of splits above η, $0 \leq \alpha_T \leq 1$ controls the shape (balance) of the tree, and $\beta_T \geq 0$ controls the size of the tree. In general, decreasing β_T increases the probability of having a tree with more terminal nodes. The treed model prior, $P(\mathcal{T})$, is computed as

$$P(\mathcal{T}) = \prod_{\eta \in \mathcal{I}} P_{split}(\eta, \mathcal{T}) \prod_{\eta \in \mathcal{L}} [1 - P_{split}(\eta, \mathcal{T})],$$

where $\mathcal{I}$ and $\mathcal{L}$ denote the set of internal nodes and terminal nodes in $\mathcal{T}$, respectively. The splitting rule probability $P_{rule}(\rho|\eta, \mathcal{T})$ is taken to be a uniform distribution on the set of available splitting dimensions and values. The prior for the set of all splitting rules is $P(\boldsymbol{\rho}|\mathcal{T}) = \prod_{\eta \in \mathcal{I}} P_{rule}(\rho|\eta, \mathcal{T})$.

To specify $P(\boldsymbol{\Theta}|\mathcal{T})$, we follow the suggestion by Chipman et al. (1998) and impose conjugate priors on the terminal node parameters and assume conditional independence of parameters across terminal nodes. Doing this allows us to analytically marginalize the terminal node parameters out from the joint posterior. The conjugate priors are specified as follows,

$$\begin{aligned}
&\boldsymbol{\beta}_\nu|\boldsymbol{\beta}_0, \boldsymbol{\phi}_\nu, \mathbf{C}, \boldsymbol{\rho}, \mathcal{T} \sim N_{p+1}(\boldsymbol{\beta}_0\,, (\phi_\nu \mathbf{C})^{-1}), \quad \phi_\nu|b_y, \boldsymbol{\rho}, \mathcal{T} \sim G(a_y\,, b_y),\\
&\mathbf{x}_\nu|\lambda_\nu, \boldsymbol{\rho}, \mathcal{T} \sim N_{m_\nu}(\mathbf{0}\,, (\lambda_\nu \mathbf{I}_{m_\nu})^{-1}),
\end{aligned}$$

where G denotes the Gamma distribution. Note that $\mathbf{x}_\nu$ is given a Gaussian distributed prior each having a separate precision. As a result, this puts a nonstationary Gaussian prior on $\mathbf{z}$. We define the hyper-priors as follows

$$\lambda_\nu | b_x, \boldsymbol{\rho}, \mathcal{T} \sim G(a_x\,, b_x) \quad \boldsymbol{\beta}_0 \sim N(\boldsymbol{\mu}, \mathbf{B}^{-1}), \quad \mathbf{C} \sim W((\varphi \mathbf{V})^{-1}, \varphi),$$
$$b_y \sim G(\tau_y, \xi_y), \quad b_x \sim G(\tau_x, \xi_x),$$

where a_x, a_y, $\boldsymbol{\mu}$, $\mathbf{B}$, φ, $\mathbf{V}$, τ_x, ξ_x, τ_y, ξ_y are constants, and W denotes the Wishart distribution. The kernel precision matrix $\mathbf{Q}_\nu$ is given a Wishart distribution prior, $\mathbf{Q}_\nu | \boldsymbol{\rho}, \mathcal{T} \sim W((\psi \mathbf{H})^{-1}, \psi)$, where ψ denotes the degrees of freedom and $\mathbf{H}$ denotes a scale matrix such that $\mathbf{H}^{-1}$ is the prior mean. Combining the priors for all the parameters with the tree priors $P(\mathcal{T})$ and $P(\boldsymbol{\rho}|\mathcal{T})$ along with the likelihood given in (6) provides all the necessary ingredients for using Reversible Jump Markov Chain Monte Carlo (RJ-MCMC) to explore the joint posterior distribution of the treed model and unknown parameters. Specifically, the Metropolis-Hastings algorithm is used to explore the posterior space of $(\mathcal{T}, \boldsymbol{\rho})$ and $\mathbf{Q}_\nu$ while Gibbs sampling is used to obtain posterior samples of the other parameters. Note that since $\mathbf{Q}_\nu$ can not be integrated out from the treed model posterior, for the *Grow* and *Prune* steps described in the next section, the proposed $\mathbf{Q}_\nu$ is drawn from its prior, so that term cancels the required Jacobian term in the acceptance ratio. A good primer on the use of RJ-MCMC is Gelman et al. (2004).

4.3 Treed model proposals

The key to implementing RJ-MCMC is the treed model proposals. We implement these proposals via the *Grow*, *Prune*, *Change* and *Swap* operations as discussed by Chipman et al. (1998). These four tree modification operations are randomly selected during each MCMC iteration. Consider the case where the *Grow* operation is chosen for the current treed model $\mathcal{T}^k$. A leaf node η at depth d_η is uniformly selected among the available candidates. Let this selected leaf node correspond to partition $\mathcal{S}_\nu$. Within this leaf node, a spatial dimension and a splitting value are uniformly selected from the available candidates. Availability is ensured when the split does not lead to empty terminal nodes. The selected leaf splits as a new parent node and is assigned a new splitting rule ρ^+. The data within this new parent node (a leaf node before splitting) is divided among its newly created children nodes according to the splitting rule. Denote this proposed treed model by

$\mathcal{T}^*$. Notice that going from $\mathcal{T}^k$ to $\mathcal{T}^*$, there is an additional parameter, ρ^+. Since ρ^+ is sampled from its prior (a uniform distribution on both the splitting dimension and values), the Jacobian term that usually exists in the acceptance ratio can be omitted. Let $\mathcal{G}$ denote the set of growable leaves of $\mathcal{T}^k$ and $\mathcal{P}$ denote the set of pruneable nodes of $\mathcal{T}^*$. Going from $\mathcal{T}^k$ to $\mathcal{T}^*$, the proposal probability is

$$q(\mathcal{T}^*, \boldsymbol{\rho}^*|\mathcal{T}^k, \boldsymbol{\rho}^k) = \frac{P_{rule}(\rho^+|\eta, \mathcal{T}^k)}{|\mathcal{G}|}, \quad \text{where} \quad \boldsymbol{\rho}^* = (\boldsymbol{\rho}^k, \rho^+).$$

Going back from $\mathcal{T}^*$ to $\mathcal{T}^k$, the proposal is simply $q(\mathcal{T}^k, \boldsymbol{\rho}^k|\mathcal{T}^*, \boldsymbol{\rho}^*) = 1/|\mathcal{P}|$. When the *Prune* operation is chosen, a parent node whose children are terminal nodes is selected uniformly from the available candidates. The children nodes are collapsed and their data are absorbed by the selected parent, which becomes a terminal node after the children nodes are removed. Since the Prune and Grow operations are counterparts of one another, the proposal distribution of Prune is just the reverse of that of the Grow operation.

When the *Change* operation is chosen, an internal node is picked uniformly from the available candidates in the current tree $\mathcal{T}^k$. While keeping the chosen splitting dimension unchanged, the splitting value is replaced by randomly sampling a new value from the available candidates. As a general rule, this new split value must not yield any empty terminal nodes in the subtree since other splitting rules in the subtree are kept fixed. After a new splitting value is obtained, data at the terminal nodes of the subtree are rearranged in order to form the new treed model $\mathcal{T}^*$. Going from $\mathcal{T}^k$ to $\mathcal{T}^*$ or vice versa, the same internal node has to be picked and the number of available split values are the same. Therefore, $q(\mathcal{T}^*, \boldsymbol{\rho}^*|\mathcal{T}^k, \boldsymbol{\rho}^k) = q(\mathcal{T}^k, \boldsymbol{\rho}^k|\mathcal{T}^*, \boldsymbol{\rho}^*)$, which can be omitted from the calculation of the acceptance ratio.

When the *Swap* operation is chosen, a parent-child pair of internal nodes are uniformly selected among the available candidates in the current tree $\mathcal{T}^k$. Then, the parent node's splitting rule is swapped with the child node's splitting rule. When the parent node splits at a different dimension than that of the child node, we have the following cases of swapping:

- If the child node is on the right, the left subtree of the parent node swaps with the left subtree of the child node.

- If the child node is on the left, the right subtree of the parent node swaps with the right subtree of the child node. See top left panel of Figure 1.

- If both children nodes have the same splitting rule, then they swap their split rules with that of the parent node's, and the right subtree of the left child node swaps with the left subtree of the right child node. See top right panel of Figure 1.

When the parent node splits at the same dimension as that of the child node's, swapping their splitting rules would yield empty terminal nodes. Instead of swapping, as noted by Gramacy (2005), a rotation scheme is used in this situation. If the child node is on the right, a left rotation is done, or vice versa. An example of a left rotation is shown in bottom left panel of Figure 1. In all cases, data at the terminal nodes are adjusted to form the new treed model $\mathcal{T}^*$. Since both the current and the proposed treed model have the same number of parent-child internal nodes pairs, the proposal is symmetric. Therefore, $q(\mathcal{T}^*, \boldsymbol{\rho}^*|\mathcal{T}^k, \boldsymbol{\rho}^k) = q(\mathcal{T}^k, \boldsymbol{\rho}^k|\mathcal{T}^*, \boldsymbol{\rho}^*)$, which can be omitted from the calculation of the acceptance ratio.

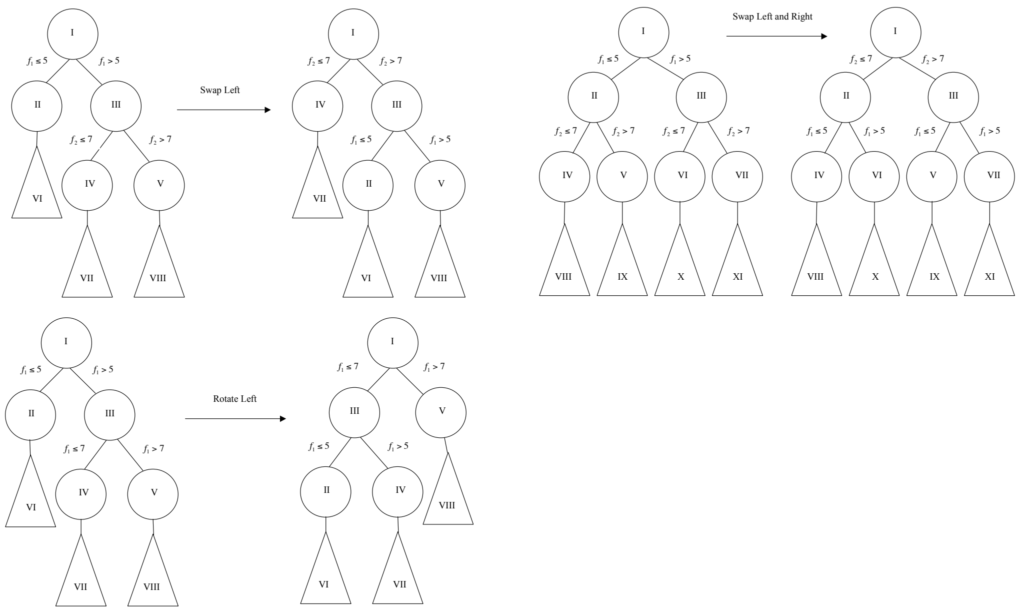


Figure 1: Examples of swap/rotate operations: left swap (*top left*), left and right swap (*top right*), and left rotate (*bottom left*)

5 Illustration

5.1 2-d Synthetic Exponential Data

Consider the following response surface from Gramacy and Lee (2009) over a $[-3,6] \times [-3,6]$ input space: $z(\mathbf{s}) = 2s_1 \exp(-\mathbf{s}^\top \mathbf{\Sigma} \mathbf{s})$, where $\mathbf{s} = (s_1, s_2)^\top$, and

$$\mathbf{\Sigma} = \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix}, \quad \text{if } s_1 < 3.5 \text{ and } s_2 < 3.5,$$

otherwise $\mathbf{\Sigma}$ is the identity matrix. This response surface is shown in Figure 2. The peak and trough of the surface are stretched towards the upper left direction, which is an example of a response surface requiring an anisotropic model. Data points $y(\mathbf{s})$ (shown at the bottom panels of Figure 2) are generated by randomly selecting 500 samples from the response and adding $N(0, 0.1)$ noise to the samples. A grid of 20×20 bases is fixed over the spatial domain as shown in the bottom right panel of Figure 2). We use the same tree prior as before with the following conjugate priors,

$$\begin{aligned}
&\phi_\nu | b_y, \boldsymbol{\rho}, \mathcal{T} \sim G(a_y = 1,\ b_y), \\
&\mathbf{x}_\nu | \lambda_\nu, \boldsymbol{\rho}, \mathcal{T} \sim N_{m_\nu}(\mathbf{0},\ (\lambda_\nu \mathbf{I}_{m_\nu})^{-1}), \quad \lambda_\nu | b_x, \boldsymbol{\rho}, \mathcal{T} \sim G(a_x = 1,\ b_x), \\
&b_y \sim G(\tau_y = 1,\ \xi_y = 1000), \quad b_x \sim G(\tau_x = 1,\ \xi_x = 1000).
\end{aligned}$$

Note that the linear term is omitted since the data has mean zero. We apply the treed model to this dataset using a Bézier ($\kappa = 3$) kernel (see the Appendix for details of a Bézier kernel), where a separate kernel precision matrix $\mathbf{Q}_\nu$ is assumed for each partition. The prior for $\mathbf{Q}_\nu$ is specified as $W\left((\psi \mathbf{H})^{-1} = \frac{1}{3}\left(\frac{3\times 9}{20-1}\right)^{-2} \mathbf{I}_2,\ \psi = 3\right)$, where $\frac{9}{20-1}$ is the spacing between any two adjacent basis points, and the degree of freedom $\psi = 3$ makes this prior fairly non-informative (ψ has to be at least the dimension of the spatial domain). A Wishart distribution is used as the proposal for $\mathbf{Q}_\nu$, where the mean is set to be the previous MCMC sample and the degrees of freedom is set equal to 100. The resulting posterior predictive surface and the corresponding 90% interval width are shown at the top and bottom left panels of Figure 3, respectively. The mean surface closely resembles the true response and the partition boundaries (*white dashed lines*) of the MAP treed model is around where the anisotropy changes. The ellipses depict the orientation of the kernel support. They have been down-scaled from their original size for visibility in the graph. The orientation of these ellipses illustrate that the treed model is able to capture anisotropy by rotating/stretching the kernels in the lower left region while keeping other regions as isotropic as possible. The estimated mean of the measurement error standard deviation is shown in the bottom right of Figure 3. Although the values are different across partitions, they are very close to the true value of 0.1. Looking at the number of partitions visited shows that majority of the models have 3 partitions while some have 4, and a small number of them have 5.

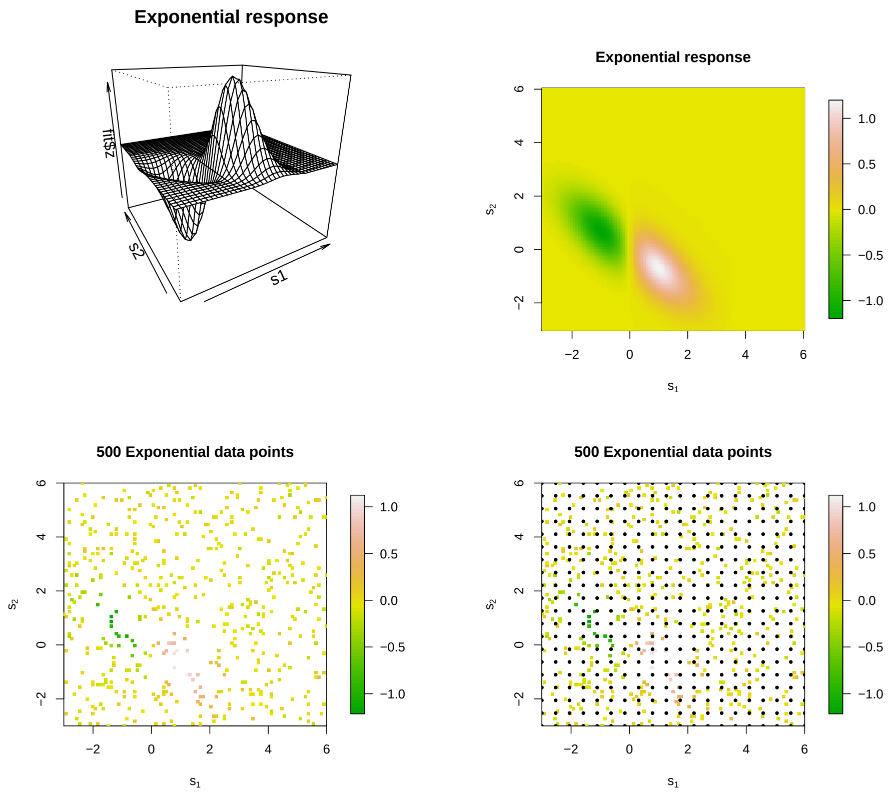


Figure 2: 2-d synthetic exponential response is shown at the top. The generated data is shown on the bottom with a (20 × 20) grid of basis points (*black solid dots*) shown on the bottom right.

5.1.1 2-d real precipitation data

Next, we apply our model to a set of precipitation data obtained from the *KriSp* R package. This dataset contains the total precipitation counts in milliliters (standardized on the square-root scale) for April 1948 recorded at 11,918 locations over the contiguous U.S. The standardization was done

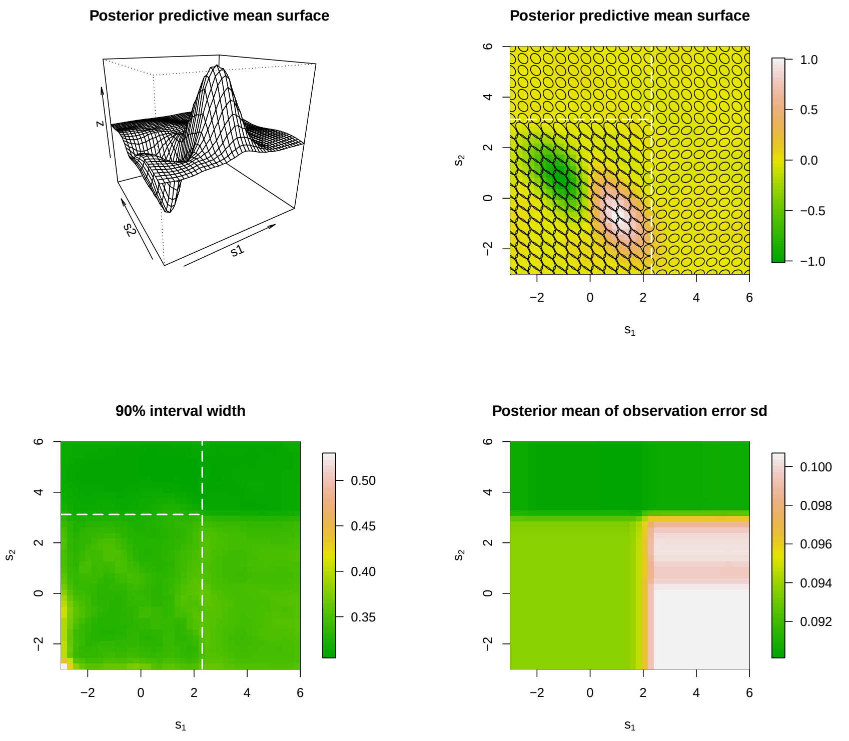


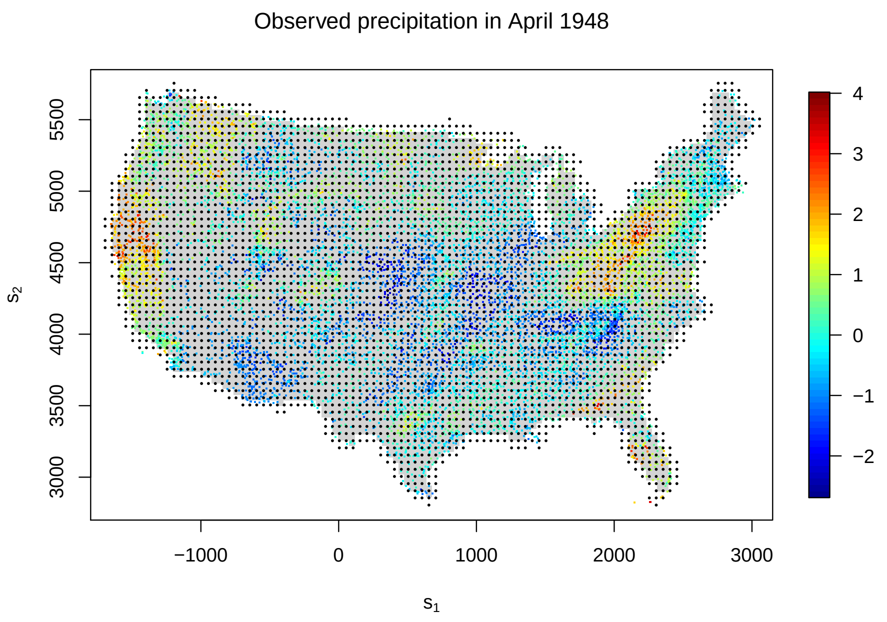
Figure 3: Fitting of 2-d synthetic exponential data based on a Bézier ($\kappa = 3$) kernel with a separate $\mathbf{Q}$ for each partition and a (20×20) grid of basis points.

(by the *KriSp* R package) to make the response more closer to a Gaussian distribution (Johns et al., 2003). The location of the data was originally represented in longitude and latitude. However, since our model assumes a planar coordinate system, the spatial representation is converted to kilometers using the Universal Transverse Mercator coordinate system. Top panel of Figure 4 shows $10,000$ of these data points superimposed by a basis grid (*black dots*) which will be used for modeling. These $10,000$ data points are randomly selected from the full dataset. The remaining data points are left out for prediction comparison. The bottom panel of Figure 4 shows the elevation over the contiguous U.S., which will be incorporated into the model as a covariate. There are $3,646$ basis points in total, and the spacing between any two adjacent basis points (along the easting or northing direction) is 50 kilometers. A Bézier ($\kappa = 0.5$) kernel is used, and the chosen value of κ is based on our experience of modeling this dataset; we found that a lower value of κ gives a better model fitting than a higher value, but improvement diminishes for values lower than 0.5. Furthermore, using the Bézier kernel allows dedicated sparse matrix computation which can speed up the model. The tree prior $P(\mathcal{T})$ has parameters $\alpha_T = 0.95$ and $\beta_T = 0.5$ as before. The other priors are specified as follows,

$$\begin{aligned}
&\boldsymbol{\beta}|\boldsymbol{\beta}_0, \mathbf{C}, \boldsymbol{\rho}, \mathcal{T} \sim N_{p+1}(\boldsymbol{\beta}_0,\ (\phi_\nu \mathbf{C})^{-1}), \quad \phi_\nu | b_y, \boldsymbol{\rho}, \mathcal{T} \sim G(a_y = 10,\ b_y),\\
&\mathbf{x}_\nu|\lambda_\nu, \boldsymbol{\rho}, \mathcal{T} \sim N_{m_\nu}(\mathbf{0},\ (\lambda_\nu \mathbf{I}_{m_\nu})^{-1}), \quad \lambda_\nu|b_x, \boldsymbol{\rho}, \mathcal{T} \sim G(a_x = 10,\ b_x),\\
&\boldsymbol{\beta}_0 \sim N(\boldsymbol{\mu} = (0,0)^\top,\ \mathbf{B}^{-1} = 0.1\mathbf{I}_2), \quad \mathbf{C} \sim W((\varphi \mathbf{V})^{-1} = 10\mathbf{I}_2,\ \varphi = 100),\\
&b_y \sim G(\tau_y = 1,\ \xi_y = 1000), \quad b_x \sim G(\tau_x = 1,\ \xi_x = 1000).\\
&\mathbf{Q}_\nu \sim W\Big((\psi \mathbf{H})^{-1} = \frac{1}{10}\Big(3 \times 50\Big)^{-2}\mathbf{I}_2,\ \psi = 10\Big),
\end{aligned}$$

Note that instead of having a separate $\boldsymbol{\beta}$ for each partition, we assume a single $\boldsymbol{\beta}$ over the entire domain. The reason is that for this particular dataset, the MCMC has convergence issues when separate $\boldsymbol{\beta}$s are used. In addition, assuming a single $\boldsymbol{\beta}$ keeps the resulting process continuous, which we believe to be a more reasonable behavior for environmental processes such as precipitation. An intercept term is also included, so $\boldsymbol{\beta}$ has dimension (2×1). Its posterior distribution can easily be re-derived in a similar fashion.

According to Chipman et al. (1998), the posterior distribution of a treed model is often multi-modal, and the MCMC is likely to get stuck in a local mode. One solution proposed by Chipman



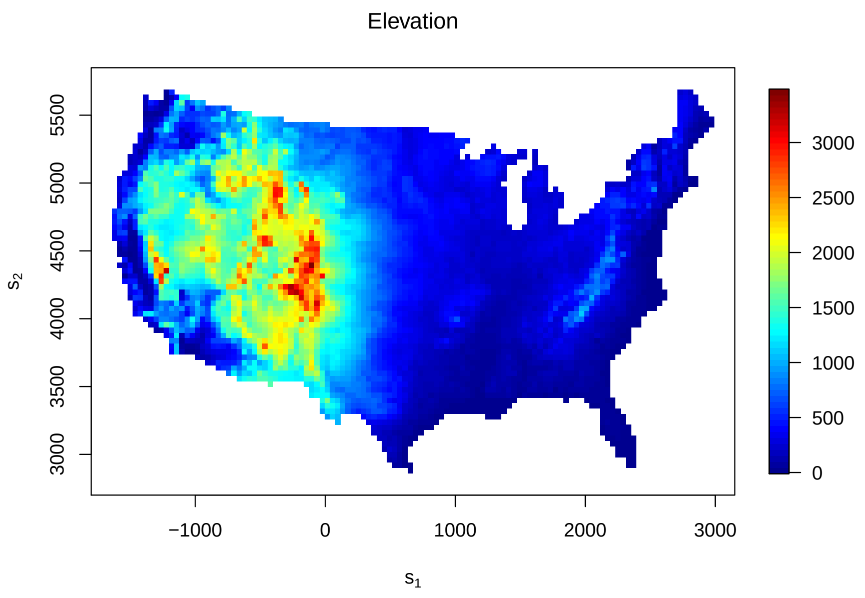
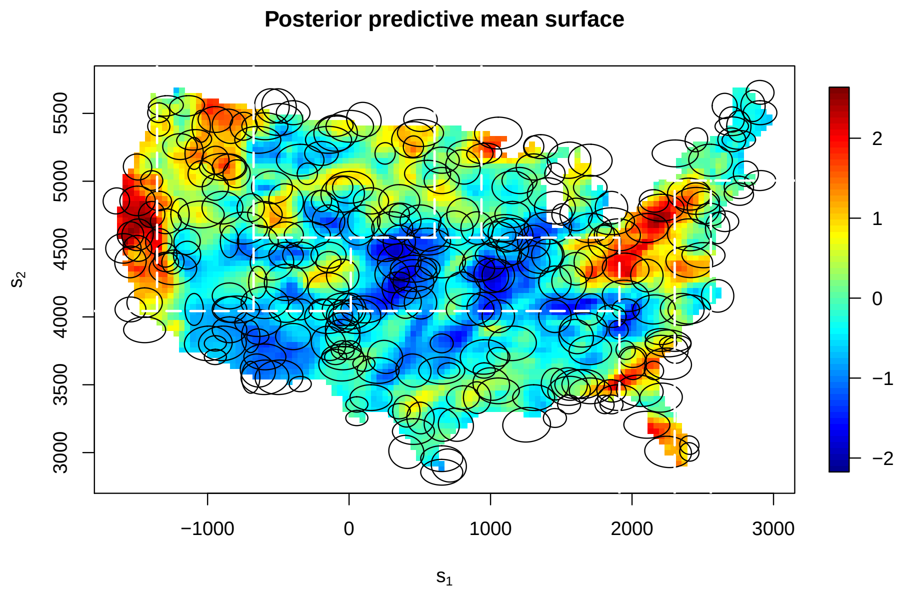


Figure 4: *Top panel*: Total precipitation count for April 1948 over the contiguous U.S. with a basis grid denoted by black dots. *Bottom*: Elevation of the contiguous U.S.

et al. (1998) is to restart the MCMC multiple times (with a different random seed) and average the different models. Having restarts might seem time consuming, however, computers nowadays have multiple cores/threads per CPU, so multiple MCMCs can be run in parallel on a single chip. In fact, we did 8 independent runs of the model for this dataset. Each run is hosted on a thread of an Intel i5 quad-core (2 threads each) processor at 2.8 GHz. Each run has 3000 iterations with the first 2000 as burn in, giving a total of 1000 posterior samples. The posterior predictive mean surface and the corresponding 90% interval width are shown in Figure 5. These surfaces are obtained by averaging the samples from all 8 independent runs of the model. The white dashed lines depict the partition boundary, which corresponds to the MAP treed model among all models visited by the 8 independent runs. The ellipses represent the support of 300 randomly selected kernels from a total of $3,646$ each having $8,000$ MCMC samples. This shows that the resulting model fitting is not only due to the nonstationary background points $\mathbf{x}$ but also as a result of varying the shape and size of the kernels across the entire domain. The number of partitions ranges from 4 to 12, however, most of the runs cluster around 8 partitions. This is a typical behavior of treed model where the MCMC tends to get stuck in a local mode. This is also why it helps to average the model over multiple runs. Although results can be improved by having more runs, the model fitting from the average of the given 8 models is sufficiently good. This can be shown by the fitted residuals given in Figure 6. Visually, the residuals appear to be randomly scattered. To find out how correlated the residuals are, we perform a test called the "Moran's I test", which is a test for spatial autocorrelation. The function we use is called *Moran.I* from the *ape* R package. The computed p-value is about 0.75, providing no evidence of a lack of spatial fit for our model.

For model comparison, we apply MARS (Multivariate Adaptive Regression Splines) (Friedman, 1991) and covariance tapering (Furrer et al., 2006) to the same dataset. As mentioned before, MARS is a nonparametric method which uses a set of bases (hinge functions) to model the process of interest. We use an interaction degree of 3 and the model fitting is given in the left panel of Figure 7. Although the computational time (less than 30 seconds) is much faster than that (about 50 hours) of our treed model, most of the spatial features are not being captured by MARS, and increasing the interaction degree has negligible effect on the result. On the other hand, covariance tapering is an interpolation method based on kriging and a compactly supported correlation function so that sparse matrix operations can be deployed to speed up computation. The results of covariance



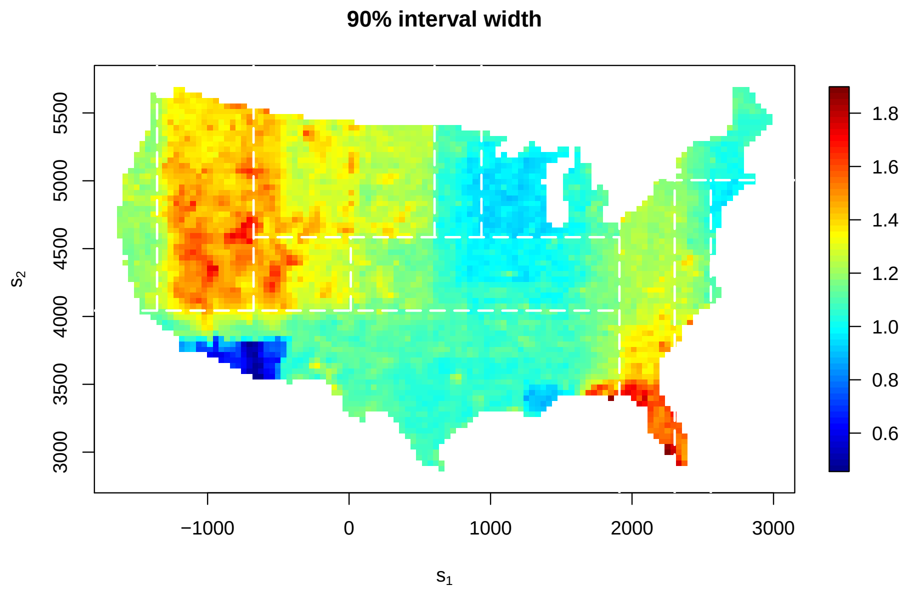


Figure 5: Posterior predictive mean surface (*top*) with the respective 90% interval width (*bottom*).

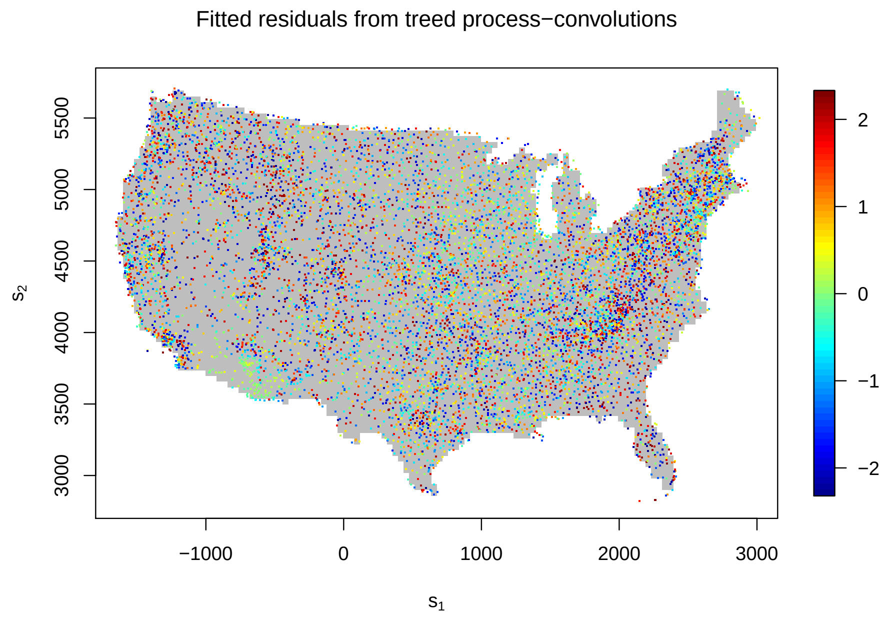


Figure 6: Residuals from fitting of precipitation data.

tapering on the precipitation dataset is provided in the right panel of Figure 7. The fitted surface from resembles that of the posterior predictive mean surface (which is very close to fitted mean surface) of our model. Since covariance tapering is an interpolation method, fitted residuals at the data locations are zero which makes comparison not meaningful. Therefore, we seek to compare prediction performance by predicting at the $1,918$ data locations that were left out from the MCMC and computing the residuals based on the corresponding observations. The mean squared error (MSE) for covariance tapering is about 0.147, which is bigger than the MSE of 0.127 for the treed model. This suggests that the treed model has better prediction performance than covariance tapering for this dataset. Other than these two models, we also tried to compare to the predictive process model by Banerjee et al. (2008). The function is named *spLM* and provided in the *spBayes* R package. We called the function based on the same set of data and basis, however, the MCMC did not finish within two weeks, so we decided to give up on this comparison. Conceptually, our model has an advantage not only in the case of heteroskedasticity, but also in situations where the process of interest is nonstationary overall but locally stationary with region-specific anisotropy. Our treed model is more robust than competing models that require knowledge of the specific regions (e.g., for applying a nonstationary Matérn correlation in the predictive process model) because it can

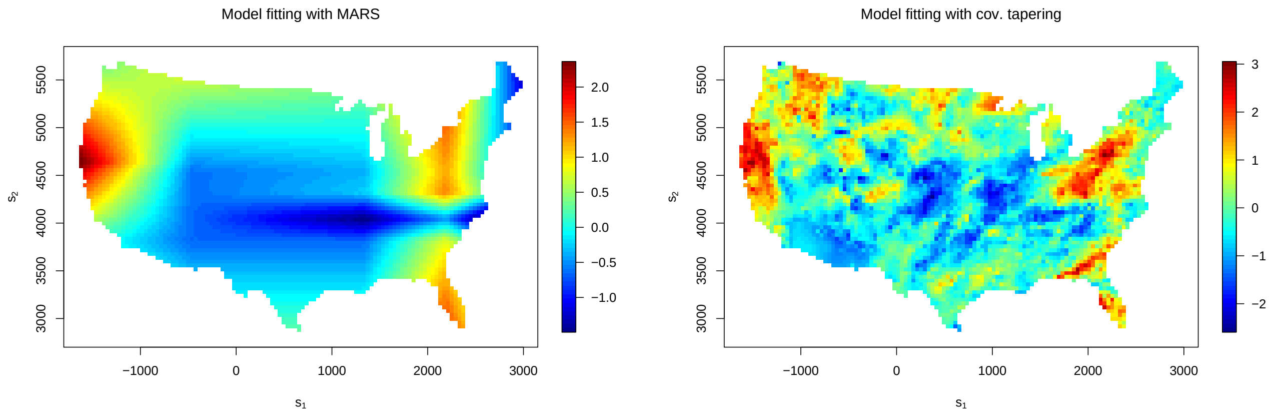


Figure 7: Model fitting from MARS (*left*) and covariance tapering (*right*).

find the regions automatically by partitioning and handles local patterns by varying the kernels across partitions.

One drawback of the treed model is that the computational expense increases due to the Metropolis-Hastings step for the set of kernel precision matrices $\{\mathbf{Q}_\nu, \nu = 1, \cdots, b\}$. In fact, the computing time of the model is only about 6 hours when the kernel precision matrix is fixed, which provides a fast approximation to the full model. Allowing the kernels to vary across partitions increases the computing time up to about 50 hours. The main cause of this increase is due to the re-construction of the kernel matrix $\mathbf{K}_\nu$ for each update of $\mathbf{Q}_\nu$. For a large sample size and number of background points such as in this problem, it requires a significant amount of memory access, which can hardly be sped up within the framework of R. One solution is to program the model in a low level language such as C/C++ so that there are more options for speeding up memory access. Compared to the predictive process model (Banerjee et al., 2008) and the standard treed GP model (Gramacy and Lee, 2008) where both are written in C/C++ but are not so speedy for this dataset, our treed model still has more room for improvement in terms of implementation.

6 Conclusion

In this paper, a nonstationary process convolution-based GP model is developed by convolving a smoothing kernel with a latent process partitioned using a method similar to that of Classification and Regression Trees. Partitioning finds regions wherein the distribution of the observations is more homogenous. The nonstationarity of this treed process convolutions GP model is based on two parts. First, allowing the variability of the background points to vary across partitions induces nonstationarity in the model at a global level since every response depends on the full set of background points over the entire domain. Second, varying the kernel precision matrix across partitions captures any local patterns that can not be explained at the global level by the background points. In addition, compactly supported kernels can be used to remove long range dependence which is often negligible in many applications, allowing the use of sparse matrix routines for more efficient computations. Together, they form a robust and computational efficient nonstationary GP model applicable for large datasets that can not be handled by the standard GP model approach.

Acknowledgments

This research was partially supported by National Science Foundation grant DMS-0906720. We thank Genetha Gray for connecting us to the application.

Appendix: Bézier Kernel

The Bézier kernel is defined as

$$k(\mathbf{u}-\mathbf{s};\mathbf{Q}) = \begin{cases} (1-D_M(\mathbf{u},\mathbf{s},\mathbf{Q})^2)^\kappa & \text{if } D_M(\mathbf{u},\mathbf{s},\mathbf{Q})<1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $D_M(\mathbf{u},\mathbf{s},\mathbf{Q}) = \sqrt{(\mathbf{u}-\mathbf{s})^\top \mathbf{Q}(\mathbf{u}-\mathbf{s})}$ denotes the Mahalanobis distance with covariance matrix $\mathbf{Q}^{-1}$. The parameter κ governs the smoothness of the resulting GP such that it is 2κ differentiable (Brenning, 2001). For isotropic Bézier kernel (when $\mathbf{Q}^{-1}$ is a diagonal matrix with identical diagonal elements), the square-root of the diagonal elements equals to the radius of the compact support. Using a compactly supported kernel ensures that $z(\mathbf{s})$ is related only to the values of z located around $\mathbf{s}$. In the matrix representation of the model (see Section 2.2), since $\mathbf{K}_{ij} = k(\mathbf{u}_j - \mathbf{s}_i)$, using a compactly supported kernel makes $\mathbf{K}$ a sparse matrix.

References

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2003), *Hierarchical Modeling and Analysis for Spatial Data*, Chapman & Hall, Boca Raton, FL.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008), "Gaussian predictive process models for large spatial data sets," *Journal of the Royal Statistical Society series B*, 70(4), 825–848.

Breiman, L., Friedman, J. H., Olshen, R., and Stone, C. (1984), "Classification and Regression Trees," *Belmont CA: Wadsworth.*

Brenning, A. (2001), "Geostatistics without stationarity assumptions within geographical information systems," *Freiberg Online Geoscience*, 6, 1–108.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998), "Bayesian CART Model Search," *Journal of the American Statistical Association*, 93, 935–948.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2002), "Bayesian Treed Models," *Machine Learning*, 48, 303–324.

Cressie, N. (1991), *Statistics for Spatial Data*, John Wiley and Sons, Inc.

Damian, D., Sampson, P., and Guttorp, P. (2001), "Bayesian estimation of semi-parametric non-stationary spatial covariance structure," *Environmetrics*, 12, 161–178.

Denison, D., Mallick, B., and Smith, A. F. M. (1998), "A Bayesian CART algorithm," *Biometrika*, 85, 363–377.

Duan, A., Guindani, M., and Gelfand, A. E. (2005), "Generalized Spatial Dirichlet Process Models," *Biometrika*, 94 (4), 809–825.

Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines," *Annals of Statistics*, 19 (1), 1–67.

Fuentes, M. and Smith, R. L. (2001), "A new class of nonstationary spatial models," Tech. rep., North Carolina State University, Department of Statistics, Raleigh, NC.

Furrer, R., Genton, M. G., and Nychka, D. (2006), “Covariance Tapering for Interpolation of Large Spatial Datasets,” *Journal of Computational and Graphical Statistics*, 15 (3), 502–523.

Gelfand, A. E., Kottas, A., and MacEachern, S. N. (2005), “Bayesian Nonparametric Spatial Modeling With Dirichlet Process Mixing,” *Journal of the American Statistical Association*, 100 (471), 1021–1035.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian Data Analysis*, Chapman & Hall.

Gramacy, R. B. (2005), “Bayesian Treed Gaussian Process Models,” Ph.D. thesis, Department of AMS, UCSC, Santa Cruz, 95060.

Gramacy, R. B. and Lee, H. K. (2008), “Bayesian Treed Gaussian Process Models with an Application to Computer Modeling,” *Journal of the American Statistical Association*, 103(483), 1119–1130.

Gramacy, R. B. and Lee, H. K. (2009), “Adaptive Design and Analyis of Supercomputer Experiments,” *Technometrics*, 51, 130–145.

Green, P. J. (1995), “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination,” *Biometrika*, 82, 711–32.

Higdon, D. (1998), “A process-convolution approach to modeling temperatures in the North Atlantic Ocean,” *Journal of Environmental and Ecological Statistics*, 5(2), 173–190.

Higdon, D. (2002), “Space and space-time modeling using process convolutions,” in *Quantitative Methods for Current Environmental Issues*, eds. C. Anderson, V. Barnett, P. Chatwin, and A. El-Shaarawi, pp. 37–54, London, Springer.

Higdon, D., Swall, J., and Kern, J. (1999), “Non-stationary Spatial Modeling,” in *Bayesian statistics 6*, p. 761768, Proceedings of the Sixth Valencia International Meeting.

Hoef, J. M. V., Cressie, N. A. C., and P., R. P. B. R. (2004), “Flexible spatial models based on the fast Fourier transform (FFT) for cokriging,” *Computnl Graph. Statist.*, 13, 265–282.

Johns, C. J., Nychka, D., Kittel, T. G., and Daly, C. (2003), "Infilling Sparse Records of Spatial Fields," *Journal of the American Statistical Association*, 98, 796–806.

Kammann, E. E. and Wand, M. P. (2003), "Geoadditive models," *Appl. Statist.*, 52, 1–18.

Kim, H.-M., Mallick, B. K., and Holmes, C. C. (2005), "Analyzing nonstationary spatial data using piecewise Gaussian processes," *Journal of the American Statistical Association*, 100, 653–668.

Lee, H. K. H., Higdon, D., Calder, C. A., and Holloman, C. H. (2005), "Efficient Models for Correlated Data via Convolutions of Intrinsic Processes," *Statistical Modelling*, 5, 53–74.

Lemos, R. T. and Sansó, B. (2009), "Spatio-Temporal Model for Mean, Anomaly and Trend Fields of North Atlantic Sea Surface Temperature," *Journal of the American Statistical Association*, 104, 5–18.

Lin, X., Wahba, G., Xiang, D., Gao, F., Klein, R., and Klein, B. (2000), "Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV," *Ann. Statist.*, 28, 1570–1600.

Paciorek, C. and Schervish, M. J. (2006), "Spatial Modelling Using a New Class of Nonstationary Covariance Functions," *Environmetrics*, 17, 483–506.

Paciorek, C. J. (2007), "Computational techniques for spatial logistic regression with large datasets," *Computnl Statist. Data Anal.*, 51, 36313653.

Sampson, P. and Guttorp, P. (1992), "Nonparametric Estimation of Nonstationary Spatial Covariance Structure," *Am. Stat. Assoc.*, 87, 108–119.

Schmidt, A. and O'Hagan, A. (2003), "Bayesian inference for non-stationary spatial covariance structure via spatial deformations," *Journal of the Royal Statistical Society, Series B*, 65, 743–758.

Stein, M. L. (1999), *Interpolation of Spatial Data*, Springer, New York, NY.

Swall, J. L. (1999), "Nonstationary Spatial Modeling Using a Process Convolution Approach," Ph.D. thesis, Duke University, Durham, NC 27708.

Wikle, C. and Cressie, N. (1999), "A dimension-reduced approach to space-time Kalman filtering," *Biometrika*, 86, 815–829.

Xia, G. and Gelfand, A. E. (2006), "Stationary process approximation for the analysis of large spatial datasets," Tech. rep., Institute of Statistics and Decision Sciences, Duke University, Durham.