Statistica Sinica **30** (2020), 2023-2050
doi:https://doi.org/10.5705/ss.202018.0232

LARGE MULTI-SCALE SPATIAL MODELING USING TREE SHRINKAGE PRIORS

Rajarshi Guhaniyogi and Bruno Sanso

University of California, Santa Cruz

Abstract: We develop a multiscale spatial kernel convolution technique that employs higher-order functions to capture fine-scale local features, and lower-order terms to capture large-scale features. To achieve parsimony, the coefficients in the proposed model are assigned a new class of "tree shrinkage prior" distributions. Tree shrinkage priors exert increasing shrinkage on the coefficients as the resolution increases, enabling them to adapt to the necessary degree of resolution at any sub-domain. In contrast to existing multiscale approaches, our approach auto-tunes the degree of resolution necessary to model a subregion in the domain, and achieves scalability by parallelizing the local updating of the parameters. The empirical performance of the proposed method is illustrated using several simulation experiments and a geostatistical analysis of sea surface temperature data from the Pacific Ocean.

Key words and phrases: Discrete kernel convolution, large spatial data, multiscale modeling, sea surface temperature, tree shrinkage prior.

1. Introduction

The ubiquity of spatially indexed data sets in various disciplines (Gelfand et al. (2010); Cressie and Wikle (2015); Banerjee, Carlin and Gelfand (2014)) has motivated researchers to develop a variety of methods and models related to spatial statistics. Most spatial applications focus on producing estimates of the mean function and the uncertainty intervals across the entire area. In many instances, spatial data exhibit global features accompanied by local variations. For example, when modeling sea surface temperature data from the Eastern Pacific, we must consider large-scale features, such as temperature off the shore of Canada being lower than that off the west coast of the United States, as well as a number of local variations, such as the variation in temperature due to upwelling along the California coast. Thus, models built upon such data should capture both large-scale spatial variations and features at the local scale. Gaussian processes offer a rich modeling framework, and are widely employed to help researchers comprehend complex spatial phenomena. However, Gaussian process likelihood computations involve matrix factorizations (e.g., Cholesky) and determinant computations for large spatial covariance matrices that have no computationally exploitable structure. This incurs an onerous computational burden for big data, and is referred to as the "Big-N" problem in spatial statistics.

We provide a brief review of the literature on big spatial data; Heaton et al. (2018) for a more comprehensive review. There are, broadly speaking, two different premises for modeling large spatial datasets: "sparsity," and "dimension-reduction." The sparse methods include covariance tapering (see, e.g., Furrer, Genton and Nychka (2006); Kaufman, Schervish and Nychka (2008); Du, Zhang and Mandrekar (2009); Shaby and Ruppert (2012)), which introduces sparsity into a covariance matrix using compactly supported covariance functions. This is effective for fast parameter estimation and interpolations of the response, but is less suited to more general inferences on residual or latent processes, owing to the exorbitantly expensive determinant computation of the sparse covariance matrix. An alternative approach introduces sparsity into an inverse of the covariance matrix (precision matrix) using conditional independence assumptions or composite likelihoods (e.g., Vecchia (1988); Rue, Martino and Chopin (2009); Stein, Chi and Welty (2004); Eidvisk et al. (2014); Datta et al. (2016); Guinness (2016)). In related literature pertaining to computer experiments, localized approximations of Gaussian process models have been proposed; see, for example, Gramacy and Apley (2015).

Dimension-reduction methods subsume the popular "low-rank" models, which express realizations of a Gaussian process as a linear combination of r basis functions (see, e.g., Higdon (2002); Stein (2007); Banerjee et al. (2008); Cressie and Johannesson (2008); Finley et al. (2009); Lemos and Sanso (2009); Guhaniyogi et al. (2011)), where $r << n$. The algorithmic cost of model fitting decreases from $O(n^3)$ to $O(nr^2 + r^3)$. However, when n is large, empirical investigations suggest that r must be fairly large to adequately approximate the parent process, in which case, nr^2 flops becomes exorbitant. Furthermore, low-rank models perform poorly when neighboring observations are strongly correlated and the spatial signal dominates the noise (Stein (2014)). Variants of the dimension-reduction methods partition large spatial data into subsets containing fewer observations, fit Gaussian processes to different subsets in parallel, and then combine the inferences from the subsets; see, for example Guhaniyogi and Banerjee (2017); Guhaniyogi et al. (2018). These methods allow an approximation of a full Gaussian process to be fit to very large data sets. Another important aspect of spatial modeling is the treatment of nonstationary covariance functions, which allows the variability to change over space. Modeling explicitly the changing covariance structure in space is a desirable feature of nonstationary processes.

Multiresolution process models have been proposed in the literature on the layering of multiple processes, usually nonstationary, at different resolutions. Here, higher-resolution layers capture small-scale behavior, while the lower resolution layers capture large-scale behavior. Several approaches model spatial surfaces at multiple scales (Liang et al. (2008); Banerjee and Finley (2007)); however, there remains a lack of research on Bayesian multiscale spatial models for big data.

Our approach combines the representation of a random field using compactly supported, multiresolution basis functions with the basis coefficients modeled using a newly developed *multiscale tree shrinkage prior*. This shrinkage prior imparts increasing shrinkage on the basis coefficients as the resolution increases. This effectively leads to a continuous analogue of selecting the number of resolutions necessary to model a given sub-domain. The framework proposed in this paper allows the higher resolutions to have a large effect in some subsets of the space, and close to no effect in other locations. This is desirable if the small-scale behavior exists in part of the field only, and if the field is nonstationary. The compactly supported basis functions and the computational strategy described in Section 3.1 yield a fast Bayesian estimation that only requires the inversion of a large number of small matrices in parallel. The proposed tree shrinkage prior, which effectively shrinks a class of parameters that have an inherent tree structure, is novel in its own right, with possible applications in statistical genomics and neuroscience, for example, identifying main effects versus interaction effects in genetic studies.

Several other important works on multiscale spatial models for big data have appeared in the literature; see, for example, Nychka et al. (2015); Katzfuss (2017), and the references therein. Although our approach shares some similarities with the recently developed LatticeKrig model (LK) (Nychka et al. (2015)), there are important differences between these two classes of models. First, whereas LK constrains the total contribution to the variance from the basis functions corresponding to the rth resolution to be of order r^{-v}, we propose using a novel shrinkage prior distribution on the basis coefficients to achieve similar goals. Both Nychka et al. (2015) and Katzfuss (2017) allow for nonstationary covariance functions, but enforce the same multiresolution structure across the entire field. In contrast, our framework allows for differential shrinkage of the basis coefficients in different sub-domains. Second, unlike LK, the proposed multiscale approach incorporates a data-dependent choice of kernel width. Third, the proposed multiscale model can be embedded naturally within a hierarchical structure in order to model non-Gaussian data; see Section 4 and the online Supplementary Material. To the best of our knowledge, LK with non-Gaussian data remains largely unexplored. A Bayesian implementation of our approach leads naturally to an effective characterization of uncertainty. Finally, the simple structure of our model means we can show both the large-support property and posterior consistency for the proposed approach. These desirable theoretical properties remain largely unexplored for most other multiresolution models.

The remainder of the paper proceeds as follows. Section 2 outlines the multiscale kernel convolution model, including the choices of knots, basis functions, basis coefficients, and priors. Section 3 discusses posterior computation strategies and computational complexity. Detailed simulation studies using Gaussian and non-Gaussian data are presented in Section 4. In Section 5 we apply the proposed model to large data on temperatures in the surface of the Pacific Ocean. Finally, Section 6 concludes the paper, and proposes a number of possible future directions. Theoretical insights and extensions to the binary regression case are provided in the online Supplementary Material.

2. Multiscale Spatial Kriging

2.1. Kernel convolutions as approximations to Gaussian processes

Let $\{w(\boldsymbol{s}) : \boldsymbol{s} \in \mathcal{D}\}$ be a spatial field of interest in the continuous domain $\mathcal{D} \subseteq \mathbb{R}^d$, for $d \in \mathbb{N}^+$. Here we focus on $d = 1, 2$. We assume the spatial process $w(\boldsymbol{s})$ follows a Gaussian process. We construct a Gaussian process $w(\boldsymbol{s})$ over $\mathcal{D}$ by convolving a continuous white noise process $u(\boldsymbol{s})$, for $\boldsymbol{s} \in \mathcal{D}$, with a smoothing kernel $K(\boldsymbol{s}, \phi)$ (ϕ might be space varying), such that $w(\boldsymbol{s}) = \int K(\boldsymbol{s}-\boldsymbol{z}, \phi)u(\boldsymbol{z})d\boldsymbol{z}$, as proposed by Higdon (2002). The resulting covariance function for $w(\boldsymbol{s})$ is fully determined by the kernel $K(\cdot)$. We can obtain a discrete approximation of this process by sampling the convolved processes on a grid. Letting $\boldsymbol{s}_1^*, \ldots, \boldsymbol{s}_J^*$ be a set of knots in $\mathcal{D}$, a discrete approximation of $w(\boldsymbol{s})$ is given by

$$\theta(\boldsymbol{s}) = \sum_{j=1}^{J} K(\boldsymbol{s} - \boldsymbol{s}_j^*, \phi)u_j, \tag{2.1}$$

where u_j denotes a basis coefficient. The J knots are typically placed in a grid in $\mathcal{D}$, though other placements of knots have appeared in the literature. By varying the kernel functions and coefficients u_j, a rich variety of processes emerge from (2.1). Following Lemos and Sanso (2009), we refer to (2.1) as a discrete convolutions of terms (DCT). When J is small, a DCT provides a computationally convenient approximation of the Gaussian process $w(\boldsymbol{s})$. However, a smaller J would provide a poor approximation, and a moderately large J would exacerbate the computational burden. These computational challenges can not be solved by the brute-force use of high-performance computing systems; thus approximations or simplifying assumptions are necessary. Using a DCT at multiple scales offers a compelling way to both reduce the computation requirements and increase the approximation accuracy. In the next few sections, we develop a multiscale DCT (MDCT) model.

2.2. Partition of domain and choice of knots

To define the MDCT, we partition $\mathcal{D}$ into mutually exclusive and exhaustive sub-domains at resolution 1. At resolution 2, each sub-domain is partitioned further into mutually exclusive and exhaustive sub-domains; this process continues up to resolution R. At the lowest level, we partition $\mathcal{D}$ into $J(1)$ subsets $\mathcal{D}_1, \ldots, \mathcal{D}_{J(1)}$. At the second level, each $\mathcal{D}_i$ undergoes P partitions such that the total number of partitions is $PJ(1)$. Similarly, at the $(r-1)$th level, the set of partitions is given by $\{\mathcal{D}_{i_1,\ldots,i_{r-1}} : i_1 \in \{1,2,\ldots,J(1)\}, i_2,\ldots,i_{r-1} \in \{1,\ldots,P\}\}$. At the rth level, each $\mathcal{D}_{i_1,\ldots,i_{r-1}}$ is partitioned into P subsets $\mathcal{D}_{i_1,\ldots,i_{r-1},1}, \ldots, \mathcal{D}_{i_1,\ldots,i_{r-1},P}$, such that $\mathcal{D}_{i_1,\ldots,i_{r-1}} = \bigcup_{i_r=1}^{P} \mathcal{D}_{i_1,\ldots,i_{r-1},i_r}, \mathcal{D}_{i_1,\ldots,i_{r-1},s} \bigcap \mathcal{D}_{i_1,\ldots,i_{r-1},s'} = \phi, \forall s \neq s'$. Therefore, the number of partitions at the rth resolution is $J(r) = P^{r-1}J(1)$. In the one-dimensional $(d=1)$, case $\mathcal{D}_{i_1,\ldots,i_{r-1},i_r}$ typically denotes an interval, and we use the bisection method to partition each interval into equal-sized subintervals for the next resolution; that is $P=2$. This naturally implies that the number of partitions at the rth level is $J(r) = 2^{r-1}J(1)$. In the two-dimensional examples, any subset at a given resolution is typically a rectangle (though other choices are also possible), each of which is divided into four equal-sized subsets; that is $P=4$ and $J(r) = 4^{r-1}J(1)$. This is a common method used to divide a domain into sub-domains; see, for example Katzfuss (2017). Partitioning a domain can be envisioned as forming a tree, with the sub-domains $\mathcal{D}_{i_1,\ldots,i_r}$ as nodes of the tree. Lower and higher resolutions correspond to the upper and lower nodes, respectively, of this tree. $\mathcal{D}_1$,. . . ,$\mathcal{D}_{J(1)}$ correspond to the uppermost nodes of the tree. Then P branches emerge from each of these nodes; leading to P^2 nodes at the second level of the tree, and so on. Indeed, for any $i_1,\ldots,i_r$, $1 \le r \le R$, we define $Subtree(\mathcal{D}_{i_1,\ldots,i_r})$ by

$$Subtree(\mathcal{D}_{i_1,\ldots,i_r}) = \{\mathcal{D}_{i_1,\ldots,i_r}\} \cup_{j=1}^{R-r-1} \{\mathcal{D}_{i_1,\ldots,i_r,i_{r+1},\ldots,i_{r+j}} : i_{r+1},\ldots,i_{r+j} \in \{1,\ldots,P\}\} \cup \{\mathcal{D}_{i_1,\ldots,i_R}\}. \quad (2.2)$$

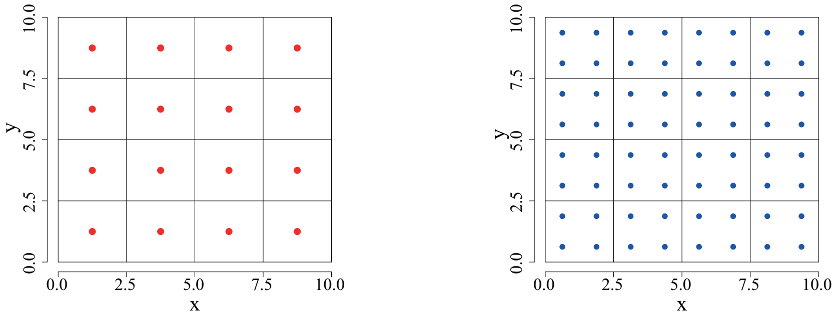
$Subtree(\mathcal{D}_{i_1,\ldots,i_r})$ consists of all sub-domains of $\mathcal{D}_{i_1,\ldots,i_r}$ higher than the rth resolution, including itself. Clearly, $Subtree(\mathcal{D}_{i_1,\ldots,i_R}) = \mathcal{D}_{i_1,\ldots,i_R}$. We define the *father* node of $\mathcal{D}_{i_1,\ldots,i_r}$ as the node $\mathcal{D}_{i_1,\ldots,i_{r-1}}$.

Defining the MDCT also requires choosing a set of knot points at every level. The knots $\boldsymbol{s}_1^1,\ldots,\boldsymbol{s}_{J(1)}^1$ at the first level are placed at the centers of $\mathcal{D}_1,\ldots,\mathcal{D}_{J(1)}$. Similarly, the knots $\boldsymbol{s}_1^r,\ldots,\boldsymbol{s}_{J(r)}^r$ are positioned at the centers of the partitions at the rth level. Technically, knots can be placed at any point in the sub-domains. However, the parallel computation for the proposed model becomes easier when the knots are placed at the centers of the intervals; see Section 3.1 which describes the computational complexity of the method when using parallelization.

There is a one-to-one correspondence between the set of knots and the set of partitions of $\mathcal{D}$. Henceforth, we refer to the $Subtree$ and $Father$ of a sub-domain with $Subtree$ and $Father$ of the knot that resides at the midpoint of that sub-domain interchangeably. For example, if $\boldsymbol{s}_j^r \in \mathcal{D}_{i_1,\ldots,i_r}$, then $Subtree(\boldsymbol{s}_j^r)$ and $Father(\boldsymbol{s}_j^r)$ are synonymous with $Subtree(\mathcal{D}_{i_1,\ldots,i_r})$ and $Father(\mathcal{D}_{i_1,\ldots,i_r})$, respectively. Note that the indexing set of knots is a bit different from the indexing set of partitions. The jth knot at the rth resolution $\boldsymbol{s}_j^r$, for $j = 1,\ldots,J(r)$, belongs to $\mathcal{D}_{i_1,\ldots,i_r}$ if $j = \sum_{l=1}^{r-1}(i_l - 1)P^{r-l} + i_r$. Using this notation, $\boldsymbol{s}_k^{r-1}$ is the father node of $\boldsymbol{s}_j^r$ iff $k = \sum_{l=1}^{r-2}(i_l - 1)P^{r-l} + i_{r-1}$; that is $k = \lfloor (j-1)/P \rfloor + 1$, where $\lfloor x \rfloor$ is the greatest integer less than x.

As examples of domain partitioning and knots for $d = 2$, let the domain of interest be $[h_1,h_2]\times[h_3,h_4]$. The first resolution divides the area into $h_x \times h_y$ equi-dimensional rectangles, with the knots placed at the center of each rectangle. The number of knots at the first resolution is $J(1) = h_x \times h_y$. At resolution 2, every rectangle at the first resolution is divided into four congruent rectangles. The knots at the second resolution are placed at the centers of these new rectangles. Clearly, the distances between two horizontally and two vertically adjacent knots are $(h_2 - h_1)/h_x$ and $(h_4 - h_3)/h_y$, respectively, at the first resolution. At the rth resolution, these distances decrease to $2^{-r+1}(h_2 - h_1)/h_x$ and $2^{-r+1}(h_4 - h_3)/h_y$, respectively.

Figure 1 shows the domain partitioning and the set of knots for two- dimensional applications. For the visual illustration, we restrict $R = 2$, $h_x = 4, h_y =$



(a) knot placement: resolution 1 (2d)

(b) knot placement: resolution 2 (2d)

Figure 1. (a) The placement of knots at resolution 1 for two dimensions; (b) the placement of knots at resolution 2 for two dimensions. For better visualization, we keep $R = 2$, and $J(1) = 16$ in two dimensions.

$4, h_1 = 0, h_3 = 0, h_2 = 10, h_4 = 10$, and $J(1) = 16$ for two-dimensional surfaces. Henceforth, the domain partitions and the placement and number of knots in each partition remains the same.

2.3. Multiscale spatial process with radial basis functions

We model the spatial effects using a MDCT with R resolutions, where the rth resolution is modeled by a DCT with kernel $K(\cdot, \cdot, \phi_r)$, knots $\boldsymbol{s}_1^r, \ldots, \boldsymbol{s}_{J(r)}^r$, and coefficients $\beta_1^r, \ldots, \beta_{J(r)}^r$, for $r = 1, \ldots, R$. The spatial surface $w(\boldsymbol{s})$ is written as $w(\boldsymbol{s}) = \sum_{r=1}^R w_r(\boldsymbol{s})$,

$$w_r(\boldsymbol{s}) = \sum_{j=1}^{J(r)} K(\boldsymbol{s}, \boldsymbol{s}_j^r, \phi_r)\beta_j^r, \tag{2.3}$$

where ϕ_r represents the scale parameter for the rth resolution. The choice of ϕ_r is discussed further later. The multiscale model represents a spatial effect using basis functions at multiple scales. The basis functions at lower resolutions have range parameters ϕ_r, that are larger than those of lower resolutions; as such, they capture variability at large distances. On the other hand, the basis functions corresponding to higher resolutions have range parameters that are small enough to describe the variability at a local level. We formally discuss the choice of basis functions and the corresponding range parameters ϕ_r below.

The choice of the kernel function $K(\cdot, \cdot, \phi_r)$ is crucial for estimating the spatial variability at multiple scales. The literature on conventional one-resolution kernel convolutions, advocates using either a Gaussian kernel or the more sophisticated Bezier kernels (Lemos and Sanso (2009); Cressie and Johannesson (2008)); which are continuous, but not differentiable for the whole family. In the multiscale literature, Nychka et al. (2015) use a Wendland kernel that is four times continuously differentiable. Let κ be a Wendland polynomial function (Wendland (2004)), supported on [0,1], and given by $\kappa(z) = (1-z)_+^{l+1}(1+(l+1)z)$, where $(1-z)_+ = (1-z)$ if $0 < z < 1$, and zero otherwise, where $l = \lfloor d/2 \rfloor + 2$. For our proposed approach, we choose a kernel function K, defined as

$$K(\boldsymbol{s}, \boldsymbol{s}_j^r, \phi_r) = \kappa\left(\frac{||\boldsymbol{s}-\boldsymbol{s}_j^r||}{\phi_r}\right) = \left(1 - \frac{||\boldsymbol{s}-\boldsymbol{s}_j^r||}{\phi_r}\right)_+^{l+1}\left[1 + (l+1)\frac{||\boldsymbol{s}-\boldsymbol{s}_j^r||}{\phi_r}\right]. \tag{2.4}$$

Geometrically, the kernel function consists of bumps centered at the node points, with the interpolation of the spatial surface at $\boldsymbol{s}$ at the rth resolution governed by knots located in $B_{\phi_r}(\boldsymbol{s})$, where $B_\nu(\boldsymbol{s})$ is the Euclidean ball of radius ν around $\boldsymbol{s}$.. Section 3.1 describes the computational advantages derived from the compact support of this kernel.

Note that κ is a Wendland polynomial function supported on $[0,1]$, and is the positive-definite, compactly-supported polynomial of minimal degree, for a given dimension d that possesses continuous derivatives up to the second order (Wendland (2004)). Theorem 1 characterizes the space of functions of the form $w_r(\boldsymbol{s})$ spanned by the basis functions $K(\boldsymbol{s}, \cdot, \phi_r)$. The proof of the Theorem 1 is given in the Supplementary Material.

Theorem 1. *Consider the reproducing kernel Hilbert space (RKHS) of the space of functions $\mathcal{H}_r = Span\,\{K(\boldsymbol{s}, \cdot, \phi_r)\}$, spanned by the kernel at the rth resolution. Then, $\mathcal{H}_r = \mathcal{S}^{d/2+3/2}(\mathcal{R}^d)$, where $\mathcal{S}^{d/2+3/2}(\mathcal{R}^d) = \{f \in L_2(\mathcal{R}^d) \cap C(\mathcal{R}^d) : \hat{f}(\cdot)(1+ ||\cdot||^2)^{(d+3)/4} \in L_2(\mathcal{R}^d)\}$, is the Sobolev space of order $d/2+3/2$, and $\hat{f}(\cdot)$ is the Fourier transform of $f(\cdot)$. $L_2(\mathcal{R}^d)$ and $C(\mathcal{R}^d)$ denote the sets of all square integrable functions and all continuous functions, respectively.*

Remark 1. Roughly speaking, this result establishes that the sample paths of $w_r(\boldsymbol{s})$ ought to provide continuously differentiable realizations of the spatial surface, a priori.

The choice of the scale parameter ϕ_r for the rth resolution follows from several considerations. First, because the kernels at lower resolutions are meant to capture long-range variability, we impose the constraint $\phi_1 > \phi_2 > \cdots > \phi_R >$

0. Second, given β_j^r, for $j = 1, \ldots, J(r)$, $r = 1, \ldots, R$, ϕ_r determines the set of knots in the neighborhood of $\boldsymbol{s}$ used to interpolate the spatial surface at $\boldsymbol{s}$.. We could keep ϕ_r as a parameter, and update it as part of the MCMC sampling. However, we found that this adds an unnecessary computational burden, with no substantial inferential advantage. Therefore, we set $\phi_r = \eta||\boldsymbol{s}_j^r - \boldsymbol{s}_{j-1}^r||$, for $\eta > 0$, where η is a tuning parameter. We do not make full Bayesian inference on η. Rather, at each step of the MCMC iteration, the posterior likelihood is maximized over a grid of η. We elaborate on this point in Section 3.1.

2.4. Multiscale spatial regression model

Our proposed multiscale spatial model typically assumes, at location $\boldsymbol{s} \in \mathcal{D}$, a response variable $y(\boldsymbol{s}) \in \mathcal{R}$, along with a $p \times 1$ vector of spatially referenced predictors $\boldsymbol{x}(\boldsymbol{s})$, which are associated through a spatial regression model, as follows:

$$y(\boldsymbol{s}) = \boldsymbol{x}(\boldsymbol{s})'\boldsymbol{\gamma} + \sum_{r=1}^{R}\sum_{j=1}^{J(r)} K(\boldsymbol{s}, \boldsymbol{s}_j^r, \phi_r)\beta_j^r + \epsilon(\boldsymbol{s}), \;\; \epsilon(\boldsymbol{s}) \sim N(0, \sigma^2), \tag{2.5}$$

where $\boldsymbol{\gamma}$ is a $p \times 1$ vector of regression coefficients. The medium- and short-range spatial variability of $y(\boldsymbol{s})$ is determined by the MDCT term, whereas $\epsilon(\boldsymbol{s})$ adds a jitter that corresponds to unexplained micro-scale variability, or measurement errors with variance σ^2.

2.5. Multiscale shrinkage prior on β_j^r

Now that the model formulation is complete, we assign prior distributions to $\beta_j^r, \boldsymbol{\gamma}$, and σ^2. Whereas the prior specification on $\boldsymbol{\gamma}$ and σ^2 is straightforward, where $\boldsymbol{\gamma}$ is assigned a noninformative prior and $\sigma^2 \sim IG(c, d)$, defining a prior distribution on β_j^r requires a bit of reflection. Note that the local variability within the spatial domain varies in relation to the sub-domains. Some regions exhibit small-scale spatial variability, while spatial variability is less prominent in other regions, which essentially do not require higher resolution terms. Mathematically, this amounts to setting $\beta_j^r = 0$ in those regions. Furthermore, it is natural to assume that if the rth resolution is deemed unnecessary to model the surface in a sub-domain, any lth resolution, for $l > r$, should be unnecessary as well, in terms of modeling for the same subregion. For $\boldsymbol{s}_j^r \in \mathcal{D}_{i_1,\ldots,i_r}$, define,

$$\mathcal{B}_{j,r}^{Subtree} = \left\{\beta_k^l : l \geq r, \boldsymbol{s}_k^l \in Subtree(\mathcal{D}_{i_1,\ldots,i_r})\right\}.$$

Thus, $\mathcal{B}_{j,r}^{Subtree}$ is the set of coefficients corresponding to basis functions centered at knots in $Subtree(\mathcal{D}_{i_1,\ldots,i_r})$. This requirement leads to condition C.

Condition C: $\beta_j^r = 0$ implies $\beta_k^l = 0$, where $\beta_k^l \in \mathcal{B}_{j,r}^{Subtree}$.

The problem of estimating β_j^r finds equivalence in the variable selection literature on high-dimensional regression where the goal is to identify predictors not related to the response or, equivalently, the predictors with coefficients equal to zero. To do so, numerous methods have been proposed, including penalized optimization methods, such as the Lasso (Tibshirani (1996)) and elastic net (Zou and Hastie (2005)), Bayesian variable selection methods and shrinkage methods. The Bayesian approach is attractive due to its probabilistic characterization of the uncertainty for regression coefficients, and the resulting predictive variability, in high dimensions.

Many Bayesian shrinkage priors have been proposed for ordinary high-dimensional regressions with a scalar/vector response on high-dimensional vector predictors; see, for example, Armagan, Dunson and Lee (2013), Hans (2009), Park and Casella (2008), Polson and Scott (2012), Carvalho, Polson and Scott (2009), and the references therein. The most popular and scalable class of high-dimensional shrinkage priors does not set predictor coefficients to zero a posteriori. Rather, these shrinkage priors are based on the principle of shrinking predictor coefficients of unimportant predictors to zero, while maintaining proper estimation and uncertainty of the important predictor coefficients. Note that a continuous analogue of Condition C would require that the prior impose greater shrinkage on coefficients in higher resolutions a priori. However, there is lack of research on such priors. Therefore, we propose the following *multiscale tree shrinkage prior* to achieve this objective:

$$\begin{aligned}&\beta_j^r \sim N(0,\alpha_j^r);\ \alpha_j^1 = \delta_1^{-1}, \alpha_j^2 = \delta_1^{-1}\delta_{j,2}^{-1}, \alpha_j^r = \alpha_{\lfloor (j-1)/P\rfloor+1}^{r-1}\delta_{j,r}^{-1};\\ &\delta_1 \sim Gamma(2,1), \delta_{j,r} \sim Gamma(c,1), c > 2,\end{aligned} \tag{2.6}$$

where $\delta_{j,r}^{-1}$ is stochastically smaller than one, implying increasing shrinkage, a priori, along a branch. In fact, $E[\beta_j^r] = 0$ and $Var[\beta_j^r] = 1/(c-1)^{r-1} \to 0$, as $r \to \infty$, a priori. Thus, the prior distribution imposes a strong a priori belief in a parsimonious model with a small number of resolutions. The proposed prior offers easy posterior updating, with closed-form conditional posterior distributions for all parameters, as is discussed in the next section.

3. Posterior Computation and Inference

This section describes the posterior computation and inference for the MDCT. The main inferential task is that of obtaining the posterior distribution of the unknown coefficients β_j^r and $\delta_{j,r}$, for $j = 1, \ldots, J(r)$ and $r = 1, \ldots, R$, $\boldsymbol{\gamma}$, and σ^2. The MDCT formulation is simple, so that all parameters allow Gibbs sampling updates. The computations for the full conditional distributions of the parameters are presented in the Supplementary Material. Samples of the posterior distribution of the parameters, obtained from the proposed sampling scheme, are used to interpolate the residual surface and perform spatial predictions. By exploiting the conditional independence between several of the parameters and the multiresolution structure of the problem, we obtain a method that makes very efficient use of computing time and memory (see Sections 3.1 and 4), and takes full advantage of distributed-memory systems with a large number of nodes (Section 3.1), and, thus, is scalable to large spatial data sets.

3.1. Distributed computation, surface interpolation, and prediction

An important advantage of the MDCT is that it facilitates distributed computation, with little communication overhead, over a large number of nodes, each processing only a small subset of the data. Section 1 in the Supplementary Material shows that posterior updating of $\boldsymbol{\gamma}, \sigma^2, \delta_{j,r}$, and δ_1 can be carried out rapidly without having to store the entire data set in centralized processing unit. The main computational difficulty comes from updating $\boldsymbol{\beta}$. Single updating of β_j^r introduces too much autocorrelation, while joint updating of $\boldsymbol{\beta}$ requires inverting a $(\sum_{r=1}^R J(r)) \times (\sum_{r=1}^R J(r))$ matrix, which is infeasible. The use of compactly supported basis functions offers an excellent solution by carefully exploiting conditional independence between blocks of $\boldsymbol{\beta}$. For $m = 1, \ldots, J(1)$, define the *neighborhood function* $\mathcal{N}(m)$ of m by $\mathcal{N}(m) = \{j : ||\boldsymbol{s}_j^1 - \boldsymbol{s}_m^1|| < 2\eta\}$. Similarly, the *neighborhood data function* is defined as $\mathcal{N}_D(m) = \{j : ||\boldsymbol{s}_j^1 - \boldsymbol{s}_m^1|| < \eta\}$. Let $\boldsymbol{\beta}_{j,r}^{Subtree}$ be a vector composed of all elements in $\mathcal{B}_{j,r}^{Subtree}$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_{1,1}^{Subtree}, \ldots, \boldsymbol{\beta}_{J(1),1}^{Subtree})'$. Exploiting the fact that knots are placed at the midpoints of every sub-domain at each resolution, and that the basis functions are compactly supported, we obtain $\boldsymbol{\beta}_{m,1}^{Subtree}|- \overset{\mathcal{L}}{=} \boldsymbol{\beta}_{m,1}^{Subtree}|\boldsymbol{y}_{\mathcal{N}_D(m)}, \boldsymbol{\beta}_{\mathcal{N}(m),1}^{Subtree}, m = 1, \ldots, J(1)$.

Algorithm 1 describes the proposed computation strategy. The computation involves $J(1)$ nodes, with the mth node storing $\{\boldsymbol{y}_{\mathcal{N}_D(m)}, \boldsymbol{X}_{\mathcal{N}_D(m)}\}$ and executing posterior updates of $\boldsymbol{\beta}_{m,1}^{Subtree}$. The main computation cost for the

Algorithm 1 Distributed computing of the posterior distribution of $\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma^2, \delta_{j,r}$

a. **No. of nodes used:** Use $J(1)$ nodes for computation.

b. **MCMC initialization:** Initialize all parameters.

c. At the tth iteration, MCMC iterates are given by $(\boldsymbol{\beta}_{m,1}^{Subtree})^{(t)}$, $m = 1, \ldots, J(1)$, $\sigma^{2(t)}, \boldsymbol{\gamma}^{(t)}$, $\delta_{j,r}^{(t)}$, $j = 2, \ldots, J(r); r = 1, \ldots, R$ and $\delta_1^{(t)}$.

d. Maximize posterior likelihood w.r.t. $\eta \in \{1, \ldots, h_\eta\}$. Compute $(\boldsymbol{y}_{\mathcal{N}_D(m)}, \boldsymbol{X}_{\mathcal{N}_D(m)})$ according to the maximized η. At the tth iteration store $(\boldsymbol{y}_{\mathcal{N}_D(m)}, \boldsymbol{X}_{\mathcal{N}_D(m)})$ in the mth node.

e. `For` $m = 1 : J(1)$ in parallel in $J(1)$ different nodes (i.) $(t+1)$ iterate of $(\boldsymbol{\beta}_{m,1}^{Subtree})^{(t+1)}$ is obtained by drawing from $\boldsymbol{\beta}_{m,1}^{Subtree} | (\boldsymbol{\beta}_{\mathcal{N}(m),1}^{Subtree})^{(t)}$.

f. `For` $m = 1 : J(1)$ in parallel in $J(1)$ different nodes (i.) Calculate $\boldsymbol{X}_m' \boldsymbol{X}_m$, $\boldsymbol{y}_m - \boldsymbol{K}_m \boldsymbol{\beta}$, where $\boldsymbol{K}_m = (K(\boldsymbol{s}, \boldsymbol{s}_1^1, \phi_1), \ldots, K(\boldsymbol{s}, \boldsymbol{s}_{J(R)}^R, \phi_R))$, $\boldsymbol{s} \in \mathcal{D}_m$.

g. Use the fact that $\sum_{m=1}^{J(1)} \boldsymbol{X}_m' \boldsymbol{X}_m = \boldsymbol{X}' \boldsymbol{X}$ and $\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\beta} = (\boldsymbol{y}_1 - \boldsymbol{K}_1 \boldsymbol{\beta}, \ldots, \boldsymbol{y}_{J(1)} - \boldsymbol{K}_{J(1)} \boldsymbol{\beta})'$ to update from the full condition of $\boldsymbol{\gamma}$.

h. Update $\delta_{j,r}^{(t+1)}$ and $\delta_1^{(t+1)}$ at the $(t+1)$th iteration.

mth node lies in computing the Cholesky decomposition of a $dim(\boldsymbol{\beta}_{\mathcal{N}(m),1}^{Subtree}) \times dim(\boldsymbol{\beta}_{\mathcal{N}(m),1}^{Subtree})$ matrix and multiplying a $dim(\mathcal{N}_D(m)) \times (\sum_{r=1}^{R} J(r))$ matrix by a vector of dimension $(\sum_{r=1}^{R} J(r))$. The computation complexity of these operations is of $O(dim(\mathcal{N}(m))^3)$ and $O(dim(\mathcal{N}_D(m)) \sum_{r=1}^{R} J(r))$, respectively. As $dim(\boldsymbol{\beta}_{\mathcal{N}(m),1}^{Subtree}) = ((2d)^R - 1)/(2d - 1)$, the computation time for the former is low. Choosing $J(1)$ sufficiently large can reduce the computation time for the latter as well. The storage complexity is also dominated by $dim(\mathcal{N}_D(m))$.

Let $\boldsymbol{s}_0$ be any location in the domain, where we seek to predict $y(\boldsymbol{s}_0)$, based on a given vector of predictors $\boldsymbol{x}(\boldsymbol{s}_0)'$. The spatial prediction at $\boldsymbol{s}_0$ proceeds from the posterior predictive distribution

$$p(\boldsymbol{y}(\boldsymbol{s}_0) \,|\, \boldsymbol{y}) = \int p(\boldsymbol{y}(\boldsymbol{s}_0) \,|\, \boldsymbol{y}, \boldsymbol{\Theta}) p(\boldsymbol{\Theta} \,|\, \boldsymbol{y}) \, d\boldsymbol{\Theta}, \tag{3.1}$$

using composition sampling, where $\boldsymbol{\Theta} = (\sigma^2, \boldsymbol{\gamma}, (\beta_j^r)_{j,r=1}^{J(r),R}, (\delta_{j,r})_{j,r=1}^{J(r),R})$. For each MCMC iteration $\{\boldsymbol{\Theta}^{(t)}\}$, for $t = 1, 2, \ldots, L$, obtained from the posterior distribution $p(\boldsymbol{\Theta} \,|\, \boldsymbol{y})$, draw $\boldsymbol{y}(\boldsymbol{s}_0)^{(t)}$ from $p(\boldsymbol{y}(\boldsymbol{s}_0) \,|\, \boldsymbol{\Theta}^{(t)})$. The resulting $\boldsymbol{y}(\boldsymbol{s}_0)^{(t)}$, for $t = 1, 2, \ldots, L$, are samples from (3.1). This is especially simple for the MDCT as $p(\boldsymbol{y}(\boldsymbol{s}_0)\,|\,\boldsymbol{\Theta})$ is a normal distribution.

For the MDCT, a full Bayesian inference on the residual spatial surface at any unobserved location $\boldsymbol{s}_0$ is trivially obtained. For each posterior sample $\{\boldsymbol{\Theta}^{(t)}\}$, for $t = 1, 2, \ldots, L$, compute $w(\boldsymbol{s}_0)_{(t)} = \sum_{r=1}^{R} \sum_{j=1}^{J(r)} K(\boldsymbol{s}_0 - \boldsymbol{s}_j^r, \phi_r)(\beta_j^r)^{(t)}$. Here $w(\boldsymbol{s}_0)_{(t)}$ denotes a sample from the posterior distribution of the residual process. Surface interpolation is then straightforward.

4. Simulation Studies

This section uses synthetic datasets to assess the proposed model's performance in terms of interpolating an unobserved residual spatial surface and predicting at new locations. First, we present a one-dimensional simulation experiment on a large data set. This experiment informs our intuitive understanding of how different resolutions capture large- and small-scale variabilities, including the advantage of using a tree shrinkage prior. Next, we present a two-dimensional example in which we compare the computation time and performance of MDCT with that of state-of-the-art and popular spatial models for big data. The methods are implemented in a nondistributed environment in R, version 3.3.1, on a 16-core processor (Intel Xeon 2.90 GHz) with 64 GB of RAM.

4.1. One-dimensional example

For the one-dimensional example, we simulated a data set of size $n = 20{,}000$ from the likelihood $N(\boldsymbol{y}|\boldsymbol{X}\boldsymbol{\gamma} + \boldsymbol{w}_0, \sigma^2)$, with a spatial function $w_0(s)$ in $[0, 10]$ given by

$$w_0(s) = \begin{cases} \sin(2\pi s)s, & \text{if } 0 \le s < 2, \\ |\sin(s-3)|^3, & \text{if } 2 \le s < 4, \\ 5|s-5|, & \text{if } 4 \le s < 6, \\ \sin(2\pi s)s, & \text{if } 6 \le s < 10. \end{cases} \tag{4.1}$$

Here, $s_1, \ldots, s_n$ are the set of spatial locations, $\boldsymbol{w}_0 = (w_0(s_1), \ldots, w_0(s_n))'$, $\boldsymbol{\gamma} = (\gamma_0, \gamma_1)$, $\boldsymbol{y} = (y(s_1), \ldots, y(s_n))'$, and $\boldsymbol{X} = (\boldsymbol{x}(s_1)' : \ldots : \boldsymbol{x}(s_n)']'$, where $\boldsymbol{x}(s_i) = (1, x(s_i))$. $x(s_i)$ is independent and identically distributed (i.i.d.) from N(0,1). A plot of the true spatial function $w_0(s)$ is provided in Figure 2. The function is piecewise differentiable, which makes the estimation challenging.

We fit the MDCT with $J(1) = 30$ to this data set. As competing methods, we implement the following:

DCT-GDP: DCT-GDP uses the same basis functions as the MDCT does, but replaces the *multiscale tree shrinkage prior* with a generalized double Pareto (GDP, Armagan, Dunson and Lee (2013)) shrinkage prior on the basis coefficients.

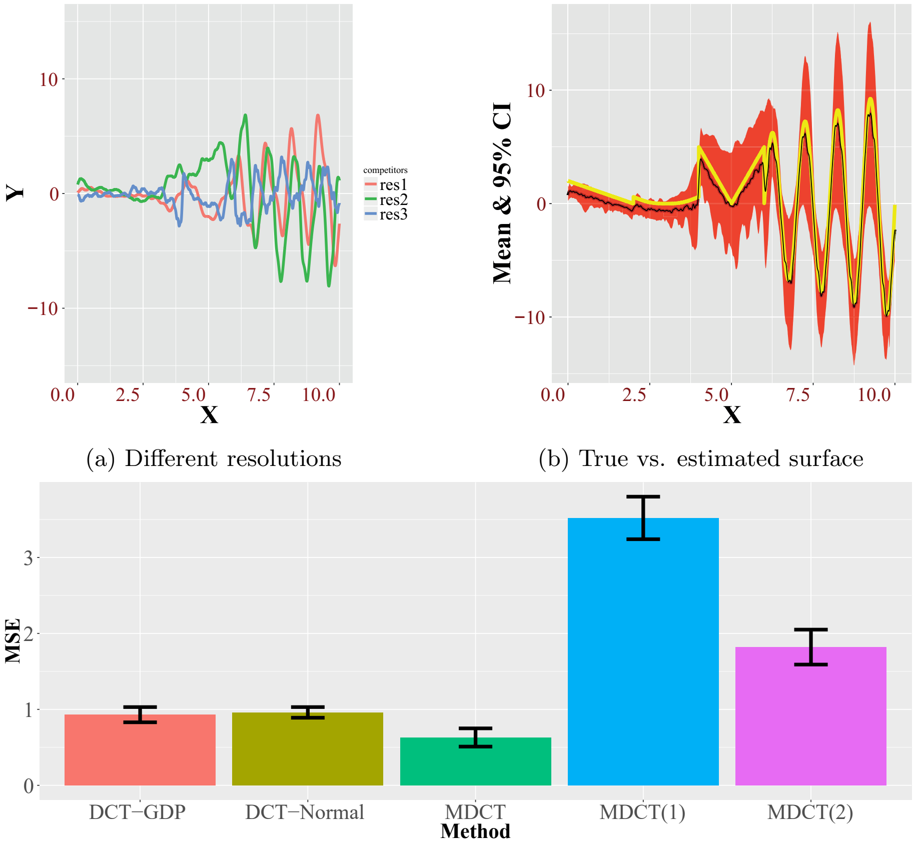
DCT-Normal: DCT-Normal again uses the same basis functions, with the prior on the basis coefficients given i.i.d. normal distributions.

DCT-GDP and DCT-Normal are used to compare the inferential advantage of the tree shrinkage prior over those of the ordinary shrinkage prior and normal prior distributions, respectively. Additionally, we fit the MDCT using one and two resolutions (MDCT(1) and MDCT(2), respectively) to assess how the choice of $R = 3$ affects the inference.

Figure 2 reveals the role played by each of the three resolutions in estimating $w_0(s)$. Resolution 1 mostly captures the positive side of the sinusoidal curve, and the negative extremities of the sinusoidal curve are mostly reconstructed by Resolution 2. Resolution 3 captures the local variability in the interval [4,10].

The inferential performance of the MDCT is evaluated by estimating the spatial surface using the mean squared error (MSE). Specifically, let $\hat{w}(s_i)$ be the posterior median of $w(s_i)$. Define the MSE $MSE = (1/n)\sum_{i=1}^{n}(\hat{w}(s_i) - w_0(s_i))^2$. Figure 2 shows the average MSE and associated standard errors over multiple simulations for the three competing models. The figures show clearly that the MDCT, with the same number of knots and the same basis functions, provides a better inference, owing to the implementation of a structured prior distribution on the basis coefficients. The computation times for the three methods are similar, with one MCMC iteration in the MDCT taking approximately 0.33 seconds to run the full-scale inference. Additionally, there appears to be a substantial improvement in terms of the MSE with increasing resolutions, though performance stabilizes after $R = 3$.

The one-dimensional exploration of the MDCT shows that multiscaling is able to capture local features succinctly, yielding a superior inference to that of a single-scale DCT, with a similar number of knots and the same basis functions. In addition, the computational advantage of the MDCT is significant, given that a full Bayesian inference can be performed using a series of local computations. Moreover, the architecture of the MDCT allows us to store subsets of data on different processors. In the next section, we compare the MDCT and several popular competitors in the context of two-dimensional spatial examples.



(a) Different resolutions

(b) True vs. estimated surface

(c) MSE

Figure 2. (a) Estimated mean function at different resolutions; (b) shows the true vs. the estimated functions for $R = 3$ resolutions. The true function is shown in bold in the middle and the estimated function is shown in overlaid. The 95% confidence bands for the estimated function are displayed by the band around it. (c) shows the MSE with associated standard errors for all competitors.

4.2. Two-dimensional example

4.2.1. Two-dimensional example with Gaussian data

This section uses two-dimensional synthetic data sets to compare the performance of the MDCT with that of popular models for large spatial data. For the sake of our exposition, the MDCT is implemented with three resolutions and 2,100 basis functions. As competitors to the MDCT we implement the following:

(1) **Modified predictive process (MPP):** The MPP (Finley et al. (2009); Banerjee et al. (2010)) is a low-rank method implemented using the package

spBayes in R.

(2) **LatticeKrig:** The LatticeKrig package in R is employed for non-Bayesian implementation of LK (Nychka et al. (2015)), with three resolutions and 12,678 basis functions.

(3) **Local approximate Gaussian process (LaGP):** The LaGP was devised by Gramacy and Apley (2015) to perform fast local-neighborhood kriging with Gaussian processes. The LaGP is not designed to provide a full-scale Bayesian inference on parameters, and is only employed to compare the predictive inference with that of the other competitors. The laGP package is implemented in R. All interpolated spatial surfaces are obtained using the R package MBA.

To illustrate the performance of the competitors, 10,500 locations $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$ are drawn uniformly from within the $[0,1] \times [0,1]$ domain. Observations are generated at these 10,500 locations from a mixture model, given by

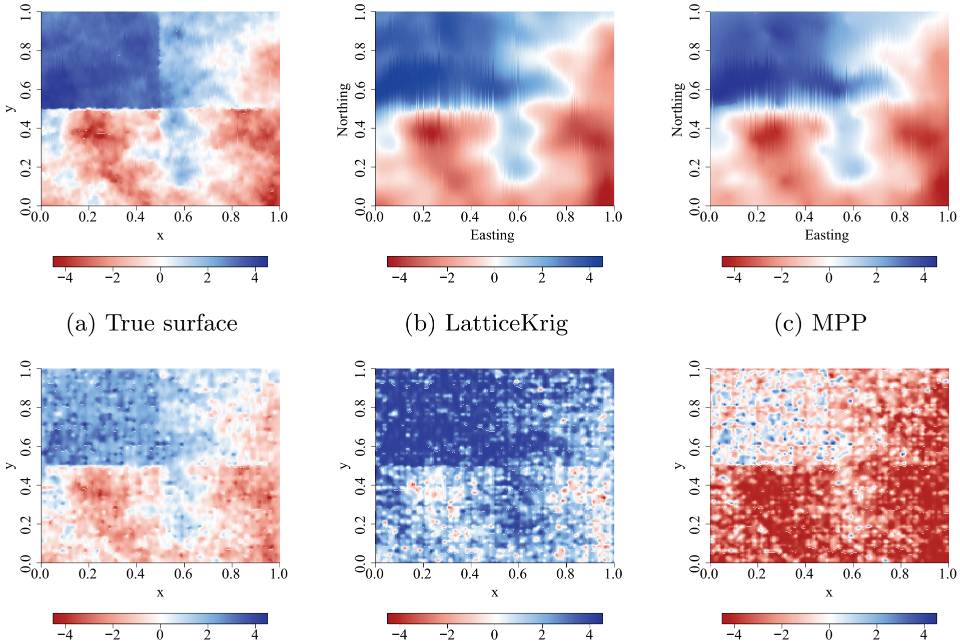
$$\begin{aligned} y(\boldsymbol{s}) = & \boldsymbol{x}(\boldsymbol{s})'\boldsymbol{\gamma} + w_1(\boldsymbol{s})I(s_1 < 0.5, s_2 < 0.5) + w_2(\boldsymbol{s})I(s_1 < 0.5, s_2 > 0.5) + w_3(\boldsymbol{s}) \\ & I(s_1 > 0.5, s_2 < 0.5) + w_4(\boldsymbol{s})I(s_1 > 0.5, s_2 > 0.5) + \epsilon(\boldsymbol{s}), \epsilon(\boldsymbol{s}) \sim N(0, \sigma^2). \end{aligned}$$

The model includes an intercept γ_0 and a predictor $x(s)$, drawn i.i.d. from $N(0,1)$, with the corresponding coefficient γ_1. We denote $\boldsymbol{\gamma} = (\gamma_0, \gamma_1)$. Here, $w_j(\boldsymbol{s})$, for $j = 1, \ldots, 4$, follows a Gaussian process with mean zero and covariance kernel $\upsilon(\boldsymbol{s}, \boldsymbol{s}', \theta_1, \theta_2, \nu))$ chosen from the popular Matern class of correlation functions given by

$$\upsilon(\boldsymbol{s}, \boldsymbol{s}', \theta_1, \theta_{2j}, \nu) = \frac{\theta_1}{2^{\nu-1}\Gamma(\nu)}(||s - s'||\theta_{2j})^{\nu}\mathcal{K}_{\nu}(||s - s'||\theta_{2j});\ \theta_{2j} > 0,\ \nu > 0, \tag{4.2}$$

where θ_{2j} and ν control the spatial decay and process smoothness, respectively, Γ is the Gamma function, and $\mathcal{K}_\nu$ is a modified Bessel function of the second kind, with order ν (Stein (2012)). We fixed $\nu = 0.5$, which reduces to the exponential covariance kernel, and generates continuous but, nondifferentiable sample paths. Additionally, $w_j(\boldsymbol{s})$ is assigned various spatial decay parameters, with $\theta_{21} = 1.5, \theta_{22} = 0.1, \theta_{23} = 1$, and $\theta_{24} = 0.5$. Of the the 10,500 observations, 10,000 are selected randomly for model fitting, and the rest are employed as a test data set to assess predictive inference.

Figure 3 presents the true data-generating surface and the estimated residual



(a) True surface (b) LatticeKrig (c) MPP

(d) MDCT mean surface (e) MDCT upper 95% CI (f) MDCT lower 95% CI

Figure 3. (a) True data-generation surface and the posterior mean residual surface from the (b) LK, (c) modified predictive process, and (d) MDCT ; (e) and (f) show the estimated 95% upper and lower quantile surfaces.

surfaces for LK, MDCT, and MPP. Here MPP shows little oversmoothing, and LK and the MDCT yield an essentially equivalent degree of precision in terms of the residual surface estimation. The 95% credible interval for the residual surface of the MDCT fits tightly around the median surface.

Next, we examine the predictive inference of the competing methods on the basis of their ability to produce accurate point predictions and predictive uncertainties. Point prediction is evaluated using the mean squared prediction error (MSPE) metric. For the Bayesian methods (i.e. the MDCT and MPP), the predictive uncertainties are characterized by the length and coverage of the 95% predictive intervals. The frequentist implementation of LK provides predictive point estimates and standard errors, and LaGP yields a posterior predictive mean and a standard error (SE) for each unobserved location. Thus, for these two competitors, the approximate 95% predictive intervals are constructed by considering the predictive point estimate $\pm 1.96\, SE$.

Figure 4 shows that the MDCT yields an MSPE that is essentially equivalent

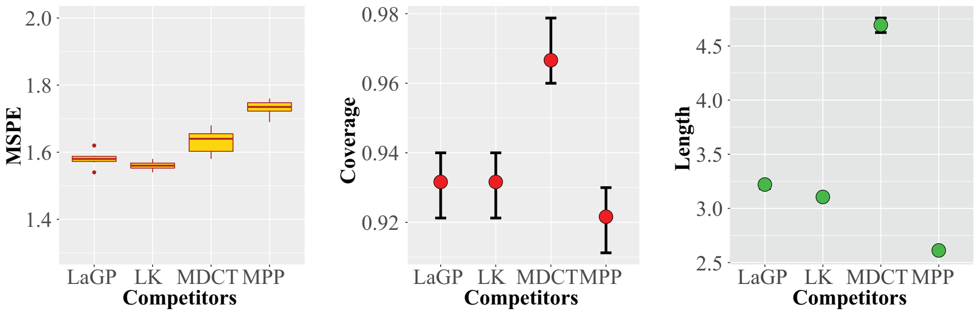


Figure 4. The left plot shows a boxplot of MSPE for each method, over a few replications. The center and right plots show the coverage and the lengths of the 95% predictive intervals, respectively, for the various methods over the same number of replications.

Table 1. Average MSPE for the MDCT with $R = 3, 4, 5$. Subscripts provide the associated standard errors over five repeated simulations.

$MDCT$	$R = 3$	$R = 4$	$R = 5$
MSPE	$1.63_{0.03}$	$1.62_{0.02}$	$1.59_{0.02}$

to those of LK and LaGP. Interestingly, the MDCT with $R = 3$ resolution shows significantly improved performance (MSPE of 1.63) over that of the MDCT with $R = 1$ (MSPE of 1.98) and $R = 2$ (MSPE of 1.85). Intuitively, we can explain this performance improvement by noting that the true surface is generated as a mixture of four region-specific Gaussian processes, with three out of four regions exhibiting significant local behavior. In terms of predictive uncertainty, the MDCT exhibits marginal over-coverage, with wider 95% credible intervals than those of the competitors. This is not surprising, given the degree of complexity embedded in the MDCT model with a large number of parameters. MPP shows a little under-coverage, with a narrower predictive intervals than those of the other methods. LK and LaGP also show competitive performance, with close to the nominal coverage, even though the 95% predictive intervals of LK correspond to a normal approximation.

To check the sensitivity with respect to the choice of R, we run our analysis with $R = 4$ and 5 and compare the results with the MSPE obtained from $R = 3$. Table 1 shows that beyond $R = 3$, the improvement in MSPE performance is not commensurate with the increase in computation cost. We found that this conclusion holds across a number of simulation studies. Therefore, $R = 3$ is used henceforth.

A few remarks are in order. Note that the MDCT and LK share similarities in terms of their multiscale structure, the only difference being the distribution of the basis coefficients. Our investigation reveals that tree shrinkage priors on basis coefficients are appropriately calibrated to yield similar point estimates to those of LK, but using far fewer basis functions. Most importantly, while the Gaussian Markov random fields (GMRF) prior distributions, used in LK, may not be conducive to parallel computation, the MDCT is able to draw a full scale Bayesian inference using a series of parallelizable local computations. Furthermore, LaGP may not be easily extendable to non-Gaussian data or to hierarchical models, as it is not model based. While the implementation of LK to non-Gaussian data is possible, but remains unexplored, the MDCT can be embedded readily into a hierarchical structure to model non-Gaussian spatial data, or to merge different sources of information, as described in the next section and in the Supplementary Material.

4.2.2. Two-dimensional example with nonGaussian data

This section briefly describes the performance of the MDCT in the presence of a nonGaussian heavy-tailed data distribution. For this simulation, 10,500 locations $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$ are drawn uniformly from a $[0,1] \times [0,1]$ domain. Observations are generated at these 10,500 locations from a mixture model, given by

$$y(\boldsymbol{s}) = \boldsymbol{x}(\boldsymbol{s})'\boldsymbol{\gamma} + w_1(\boldsymbol{s}) I(s_1 < 0.5) + w_2(\boldsymbol{s}) I(s_1 > 0.5) + \epsilon(\boldsymbol{s}),\ \ \epsilon(\boldsymbol{s}) \sim ST_2(0, \sigma^2),$$

where $ST_2(0, \sigma^2)$ denotes a Student-t distribution with two degrees of freedom and scaling parameter σ. Here, $w_1(\boldsymbol{s})$ and $w_2(\boldsymbol{s})$ are independent Gaussian processes with exponential covariance functions that have range parameters of 1.5 and 0.3, respectively. These simulations mimic the possible behavior of a random field from a variable that exhibits a distribution with long tails, over an area with sharp geographical boundaries, such as a coastline, a river, or mountain ridge. For these data, we fit an MDCT, where the error follows a Student's-t distribution,

$$y_i = \boldsymbol{x}(\boldsymbol{s}_i)'\boldsymbol{\gamma} + \sum_{r=1}^{R} \sum_{j=1}^{J(r)} K(\boldsymbol{s}, \boldsymbol{s}_j^r, \phi_r) \beta_j^r + \epsilon_i,\ \ \epsilon_i \overset{i.i.d.}{\sim} ST_2(0, \sigma^2),\ i = 1, \ldots, n. \tag{4.3}$$

Note that the Student's-t distribution can be represented as a scale mixture of normals. We exploit this fact to derive an efficient Gibbs sampler for the MDCT in Equation (4.3). Details of the posterior computations are presented in the

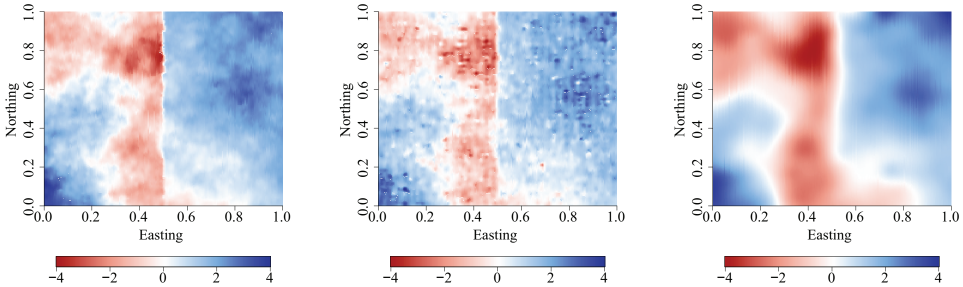


Figure 5. Left to right: True surface, median posterior surface of the MDCT, mean surface of LK, respectively.

Table 2. MSPE, length, and coverage of 95% predictive intervals of the MDCT, LaGP and LK.

	MDCT	LaGP	LK
MSPE	1.43	1.54	1.49
Length of 95% PI	9.93	6.37	5.42
Coverage of 95% PI	0.98	0.92	0.92

Supplementary Material.

In the absence of an open source implementation of LK and LaGP for non-Gaussian data, we fit the ordinary LK and LaGP to assess the performance of the MDCT. Figure 5 shows the estimated mean residual surface for the MDCT and LK, and the true surface. As expected, the MDCT is able to identify local spatial variation in the surface. Table 2 displays the MSPE, coverage, and length of the 95% predictive intervals for the three methods. Here, LaGP and LK, fitted using the normal error assumption, and the results show some under-coverage with predictive intervals narrower than those of the MDCT. In fact, the MDCT shows wider predictive intervals with little over-coverage. The MDCT slightly outperforms LK and LaGP in terms of the MSPE. The results have two important implications. First, the implementation of the MDCT can be straightforward and accurate, even with nonGaussian errors. Second, and most importantly, the Bayesian implementation of the MDCT under nonGaussian errors yields inferential advantages over its competitors. An implementation of the MDCT using a binary spatial model is given in the Supplementary Material.

Computation time: The MDCT in this example takes approximately 3.07 seconds per iteration for the non-optimized, non-parallel `R` implementation, whereas the MPP implemented in `C++` takes close to 7.2 seconds to run one MCMC iteration. However, MPP estimates the basis functions, whereas the MDCT assumes a fixed form, using the empirical Bayes estimate of η at every iteration. Here, a more elaborate inference on the kernel parameters of the MDCT might improve the predictive performance of the model. However, this would come at the cost of increased computational complexity; thus, the benefits of such an extension are not justified. In this example, the MDCT is implemented for $J(1) = 100$. To examine how the computation time of the MDCT varies in relation to that of MPP with changing n and $J(1)$, we implement both MPP and the MDCT with $J(1) = 5^2$, and 10^2 for different sample sizes. Figure 6 reports the computation times for the various methods using the `R` function `Sys.time`. Note that the MDCT can be implemented either by sequentially updating $J(1)$ blocks of parameters, or by updating these $J(1)$ blocks independently in parallel in $J(1)$ nodes. Thus, the figure shows the computation time for both parallel and nonparallel implementations of the MDCT model. Clearly, the computation time for the MDCT increases linearly with n for both cases, and the MDCT with $J(1) = 5^2$ is between four and five times faster than with $J(1) = 10^2$. The increase in computation time is due to the sequential updating of the parameters in the $J(1)$ blocks. However, with a proper parallelized implementation of the MDCT, the increase in computation time from $J(1) = 5^2$ to $J(1) = 10^2$ may be minimal. We find that a practical implementation of MPP becomes prohibitive, owing to its significant memory requirement and computation time for n above 100,000. Note that the nonBayesian implementations of LK and LaGP draw inferences for a point estimate within a few minutes. In summary, 2D simulation examples comprehensively establish the MDCT as an effective tool for fast Bayesian implementations of large-scale spatial data.

5. Analysis of Sea Surface Temperature Data

Being able to describe the evolution and dynamics of the oceans' temperature is a key component of the study of the Earth's climate. Historical records of ocean data have been collected to better understand the properties of water masses and their changes over time. They are also used to assess, initialize, and constrain numerical models of the Earth's climate. Increasingly, sophisticated climatological research requires not only a description of the mean state and the relevant trends in ocean data, but also a careful quantification of the data variability at different spatial and temporal scales. A number of works have addressed this issue; see, for example, Higdon (1998), Lemos and Sanso (2009),

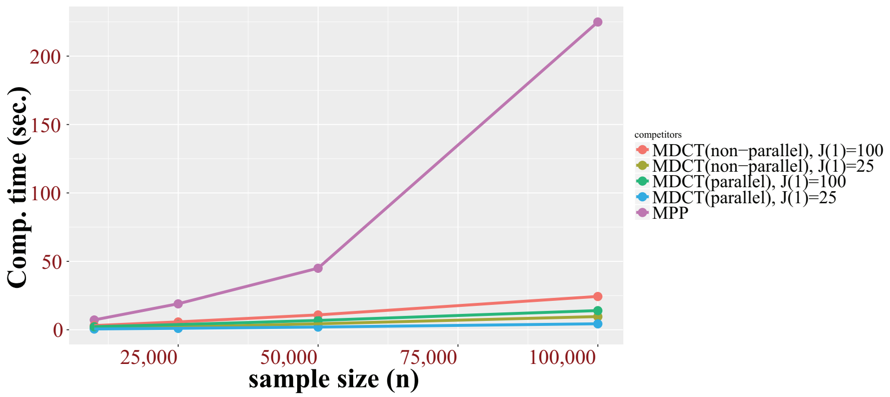


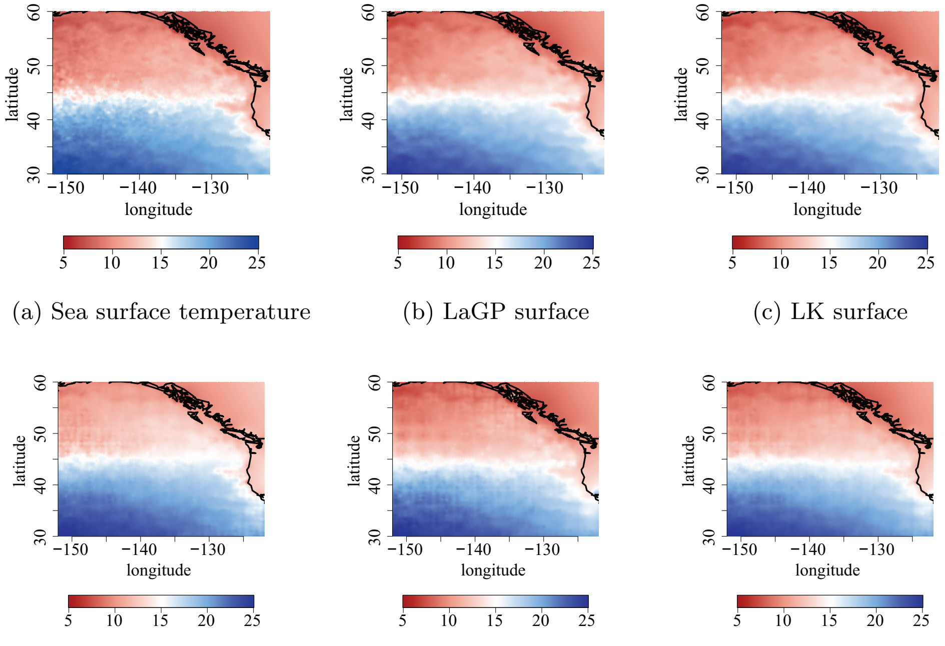
Figure 6. Computation time for MPP with 200 knots, and the MDCT with $J(1) = 25$ and $J(1) = 100$. Computation times per MCMC iteration are presented for both the MDCT and MPP.

Lemos and Sanso (2006), Berliner, Wikle and Cressie (2000), and Wikle and Holan (2011).

We consider the problem of capturing the spatial trend and characterizing the uncertainties in the sea surface temperature (SST) off the west coast of the United States, Canada, and Alaska between 30° and 60° N latitude, and 122° and 152° W longitude. The data set is obtained from the NODC World Ocean Database 2016, and we use the data collected for October. Note that, for this example, we ignore the temporal component. We screen the data to ensure quality control and then choose a random subset of 113,412 spatial observations over the domain of interest. Of the total observations, about 90%, (or 100,000) observations are used for model fitting; the remainder are used for prediction. We replicate this procedure five times to eliminate any chance factor in our analysis. The domain of interest is sufficiently large to allow considerable spatial variation in SST from north to south, and provides an important first step to extend these models for analyses of the SST database at a global scale.

The plot of the SST are shown in Figure 7a. The data show a clear decreasing SST trend with increasing latitude. Consequently, we add latitude and longitude as linear predictors to explain the long-range directional variability in the SST. We fitted a nonspatial model with latitude and longitude as linear predictors using ordinary least squares (OLS) method, and find spatial dependence with no obvious pattern of anisotropy. Thus the MDCT model with latitude and longitude as predictors seems appropriate for these data.

The proposed MDCT model for the training data uses $R = 3$ resolutions,



(a) Sea surface temperature (b) LaGP surface (c) LK surface

(d) MDCT lower 95% PI (e) MDCT upper 95% PI (f) MDCT surface

Figure 7. (a) SST in October 2016 for a portion of the North Pacific. Panels (b), (c), and (f) show the estimated mean predictive surfaces for the three competing models. Figures (d) and (e) present the pointwise predictive bands for the MDCT. Temperatures are shown in degrees centigrade.

with the first resolution having $J(1) = 100$ knots. To minimize edge effects, some knots are also kept on land. We implement Algorithm 1 and run it for 2,000 iterations, finding that $\eta = 1$ appears significantly often among the iterations. Thus, to reduce unnecessary storage complexity and to speed up the computations for a dataset of this scale, we run the remaining iterations with $\eta = 1$. The model is then run for 5,000 additional iterations. Convergence diagnostics are performed using the `coda` package in `R`, which indicates that 2,000 iterations are sufficient as a burn-in to achieve practical convergence. As competitors to the MDCT, we fitted LK and LaGP to the data. MPP is computationally prohibitive for the size of the data set and is omitted from the comparison.

The predictive power of the proposed model, along with that of its competitors, is assessed based on the MSPE, coverage, and length of 95% predictive intervals. The nonspatial model and MDCT yield an MSPE of 1.34 and 0.18 respectively. The significant improvement in the MSPE is due to the inclusion of

Table 3. MSPE, length, and coverage of 95% predictive intervals of MDCT, MDCT(1), MDCT(2), LaGP, and LK.

	MDCT	MDCT(1)	MDCT(2)	laGP	LatticeKrig
MSPE	0.18	0.52	0.36	0.11	0.10
Length of 95% PI	2.49	2.38	2.42	1.26	1.41
Coverage of 95% PI	0.98	0.95	0.97	0.93	0.93

a spatial structure, as is evident from Table 3. In fact, there is a strong spatial dependence in the field that can not be explained by a linear effect of longitude and latitude. From the results in Table 3 we observe that LaGP exhibits slightly better predictive performance than that of the MDCT. The smallest MSPE in the table corresponds to LK, fitted with $R = 3$ resolutions. Overall, the MDCT with $R = 3$ resolutions is competitive in terms of predictive inference. Importantly, even with non-parallel implementation, the MDCT takes about 26 seconds to run one iteration. As shown in Figure 6, the computation time can be reduced multiple times using an efficient parallel implementation. In contrast, even the frequentist implementation of LK takes about two hours. Thus, the MDCT with $R = 3$ resolutions outperforms MDCT(1) and MDCT(2). However, fitting the MDCT beyond $R = 3$ unnecessarily exacerbates the computational burden, without a significant improvement in inferential and predictive performance.

6. Conclusion

We propose a novel multiscale kriging model for spatial data sets. The model represents the unknown spatial surface as a sum of processes at different scales, and is able to approximate a broad class of spatial processes with various degrees of smoothness. A key component of our multiscale model is the kernel convolution, with its compactly supported kernel of minimal degree and knots placed in a regular grid at every resolution. Theoretically, this allows us to characterize completely the space of functions generated from the multiscale spatial model. Another important contribution is that we propose a new class of *multiscale tree shrinkage prior* distributions for the basis coefficients. The construction of a tree shrinkage prior is introduced under the assumption that, as the model moves to higher resolutions, increasing numbers of basis coefficients become irrelevant.

In addition to the important methodological and theoretical contributions of this study, an equally important contribution is related to computational efficiency for large data sets. The research on multiscale spatial models is largely motivated by the need to build complex and flexible spatial models that provide accurate spatial inferences and predictions for massive datasets, together with rapid Bayesian computations. The compactly supported kernel and the multiscale shrinkage priors satisfy those desiderata, providing efficient MCMC sample-based inference.

Note that the current framework for multiscale Bayesian modeling of spatial data sets can be extended readily to a spatio-temporal setting. Additionally, the recent idea of spatial meta kriging (SMK) (Guhaniyogi and Banerjee (2017)) allows scalability by fitting a spatial model independently on partitions (disjoint subsets) of big data, after which, the inferences from the subsets are combined. The proposed multiscale framework is able to scale up to approximately half a million spatial locations, but may struggle with tens of millions of locations. If we have resources to run an MDCT on, say, H different subsets with n data points each, then SMK can yield full Bayesian inference on nH locations. Finally, we have proposed to use a rectangular partition of the domain. There is a scope for future research to examine how adaptive partitioning of the domain can be implemented using techniques such as the Voronoi tesselation. Adaptive partitioning with an appropriate placement of knots might significantly reduce the number of knots required to yield an accurate inference. We leave these topics to future research.

Supplementary Material

The online supplementary material includes the following:

1. Posterior computation for the MDCT with Gaussian model.

2. Posterior computation for the MDCT with nonGaussian data.

3. Two-dimensional example of the MDCT with binary spatial data.

4. Theoretical properties.

Acknowledgments

The first author was partially supported by the Office of Naval Research, award no. N00014-18-2741. The second author was partially supported by the National Science Foundation, grant DMS-1513076.

Appendix

Proof of Theorem 1:

Use the fact that κ is a compactly supported polynomial of minimal degree for two dimensions that possesses continuous derivatives upto second order. By Theorem 10.10 and 10.35 in Wendland (2004), we obtain that the Fourier transform of κ, denoted by $\hat{\kappa}$ satisfies $c_1(1+||\omega||_2)^{-d-3} \leq \hat{\kappa}(\omega) \leq c_2(1+||\omega||_2)^{-d-3}$, for some $c_1, c_2 > 0$. The result now follows using Corollary 10.13 in Wendland (2004).

References

Armagan, A., Dunson, D. B. and Lee, J. (2013). Generalized double pareto shrinkage. *Statistica Sinica* **23**, 119–143.

Banerjee, S., Carlin, B. P. and Gelfand, A. (2014). *Hierarchical Modeling and Analysis for Spatial Data.* CRC Press, Boca Raton, FL.

Banerjee, S., Gelfand, A. E., Finley, A. O. and Sang, H. (2008). Gaussian predictive process models for large spatial datasets. *Journal of Royal Statistical Society: Series B (Statistical Methodology)* **70**, 825–848.

Banerjee, S. and Finley, A. O. (2007). Bayesian multi-resolution modelling for spatially replicated datasets with application to forest biomass data. *Journal of Statistical Planning and Inference* **137**, 3193–3205.

Banerjee, S., Finley, A.O., Waldmann, P. and Ericsson, T. (2010). Hierarchical spatial process models for multiple traits in large genetic trials. *Journal of the American Statistical Association* **105**, 506-521.

Berliner, L. M., Wikle, C. K. and Cressie, N. A. C. (2000). Long-lead prediction of Pacific SSTs via Bayesian dynamic modeling. *Journal of Climate* **13**, 3953–3968.

Carvalho, C. M., Polson, N. G. and Scott, J. G. (2009). Handling sparsity via the horseshoe. *AISTAT* **5**, 73–80.

Cressie, N. and Johannesson, G. (2008). Fixed rank kriging for very large spatial datasets. *Journal of the American Statistical Association* **70**, 209–226.

Cressie, N. A. C. and Wikle, C. K. (2015). *Statistics for Spatio-Temporal Data.* John Willey & Sons, Hoboken, NJ.

Datta, A., Banerjee, S., Finley, A. O. and Gelfand, A. E. (2016). Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association* **111**, 800–812.

Du, J., Zhang, H. and Mandrekar, V. (2009). Fixed-domain asymptotic properties of tapered maximum likelihood estimators. *The Annals of Statistics* **37**, 3330–3361.

Eidvisk, J., Shaby, B. A., Reich, B. J., Wheeler, M. and Niemi, J. (2014). Estimation and prediction in spatial models with block composite likelihoods. *Journal of Computation and Graphical Statistics* **23**, 295–315.

Finley, A. O., Banerjee, S., Waldmann, P. and Ericsson, T. (2009). Hierarchical spatial modeling of additive and dominance genetic variance for large spatial trial datasets. *Biometrics* **65**, 441–451.

Furrer, R., Genton, M. G. and Nychka, D. (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computation and Graphical Statistics* **15**, 502–523.

Gelfand, A. E., Diggle, P., Guttorp, P. and Fuentes, M. (2010). *Handbook of Spatial Statistics.*

CRC Press, Boca Raton, FL.

Gramacy, R. B. and Apley, D. W. (2015). Local Gaussian process approximation for large computer experiments. *Journal of Computation and Graphical Statistics* **24**, 561–578.

Guhaniyogi, R. and Banerjee, S. (2017). Meta kriging: scalable Bayesian modeling and inference for large spatial datasets. *Technometrics (forthcoming)*. `https://doi.org/10.1080/00401706.2018.1437474`.

Guhaniyogi, R., Finley, A. O., Banerjee, S. and Gelfand, A. E. (2011). Adaptive Gaussian predictive process models for large spatial datasets. *Environmetrics* **22**, 997–1007.

Guhaniyogi, R., Li, C., Savitsky, T. D. and Srivastava, S. (2018). A divide-and-conquer Bayesian approach to large-scale kriging. *arXiv preprint arXiv:1712.09767*.

Guinness, J. (2016). Permutation methods for sharpening Gaussian process approximations. *arXiv preprint arXiv:1609.05372*.

Hans, C. (2009). Bayesian lasso regression. *Biometrika* **96**, 835–845.

Higdon, D. (1998). A process-convolution approach to modelling temperatures in the north Atlantic ocean. *Environmental and Ecological Statistics* **5**, 173–190.

Higdon, D. (2002). Space and space-time modeling using process convolutions. *Quanti-Tative Methods for Current Environmental Issues, Springer*, 37–56.

Heaton, M., Datta, A., Finley, A., Furrer, R., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., Lindgren, F., Nychka, D. W., Sun, F. and Mangion, A. Z. (2018). Methods for analyzing large spatial data: A review and comparison. *arXiv preprint arXiv:1710.05013*.

Katzfuss, M. (2017). A multi-resolution approximation for massive spatial datasets. *Journal of the American Statistical Association* **112**, 201–214.

Kaufman, C. G., Schervish, M. J. and Nychka, D. W. (2008). Covariance tapering for likelihood-based estimation in large spatial datasets. *Journal of the American Statistical Association* **103**, 1545–1555.

Liang, S., Banerjee, S., Bushhouse, S., Finley, A. and Carlin, B. P. (2008). Hierarchical multiresolution approaches for dense point-level breast cancer treatment data. *Computational Statistics & Data Analysis* **52**, 2650-–2668.

Lemos, R. T. and Sanso, B. (2006). Spatio-temporal variability of ocean temperature in the Portugal current system. *Journal of Geophysical Research: Oceans* **111**.

Lemos, R. T. and Sanso, B. (2009). A spatio-temporal model for mean, anomaly, and trend fields of north Atlantic sea surface temperature. *Journal of the American Statistical Association* **104**, 5–18.

Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F. and Sain, S. (2015). A multiresolution Gaussian process model for the analysis of large spatial datasets. *Journal of Computation and Graphical Statistics* **24**, 579–599.

Park, T. and Casella, G. (2008). The Bayesian Lasso. *Journal of the American Statistical Association* **103**, 681–686.

Polson, N. G. and Scott, J. G. (2012). Local shrinkage rules, Levy processes and regularized regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **74**, 287–311.

Rue, H., Martino, S. and Chopin, N. (2009). Approximate bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **71**, 319–392.

Shaby, B. and Ruppert, D. (2012). Tapered covariance: Bayesian estimation and asymptotics. *Journal of Computation and Graphical Statistics* **21**, 433–452.

Stein, M. L. (2007). Spatial variation of total column ozone on a global scale. *The Annals of Applied Statistics* **1**, 191–210.

Stein, M. L. (2012). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science and Business Media, New York.

Stein, M. L. (2014). Limitations on low rank approximations for covariance matrices of spatial data. *Spatial Statistics* **8**, 1–19.

Stein, M. L., Chi, Z. and Welty, L. J. (2004). Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **66**, 275–296.

Tibshirani, R. (2014). Regression shrinkage and selection via the lasso *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **58**, 267–288.

Vecchia, A. V. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **50**, 297–312.

Wendland, H. (2004). *Scattered Data Approximation* **17**. Cambridge University Press.

Wikle, C. K. and Holan, S. H. (2011). Polynomial nonlinear spatio-temporal integro-difference equation models. *Journal of Time Series Analysis*.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**, 301–320.

Department of Statistics, University of California, Santa Cruz, CA 95064, USA.

E-mail: rguhaniy@ucsc.edu

Department of Statistics, University of California, Santa Cruz, CA 95064, USA.

E-mail: bruno@soe.ucsc.edu

(Received September 2017; accepted December 2018)