Statistica Sinica Preprint No: SS-2018-0232	
Title	Large Multi-scale Spatial Modeling Using Tree Shrinkage Priors
Manuscript ID	SS-2018-0232
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202018.0232
Complete List of Authors	Rajarshi Guhaniyogi and Bruno Sanso
Corresponding Author	Rajarshi Guhaniyogi
E-mail	rguhaniy@ucsc.edu
Notice: Accepted version subject to English editing.	

Large Multi-scale Spatial Modeling Using Tree Shrinkage Priors

Rajarshi Guhaniyogi and Bruno Sanso

Department of Applied Mathematics and Statistics,

University of California, Santa Cruz, CA 95064,

correspondence email: rguhaniy@ucsc.edu

Abstract: We develop a multiscale spatial kernel convolution technique with 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 multiscale kernel convolution model is assigned a new class of "tree shrinkage prior" distributions. Tree shrinkage priors exert increasing shrinkage on the coefficients as resolution grows so as to adapt to the necessary degree of resolution at any sub-domain. In contrast to the existing multiscale approaches, the proposed approach auto-tunes the degree of resolution necessary to model a subregion in the domain, achieves scalability by suitable parallelization of local updating of parameters. Empirical performances are illustrated using several simulation experiments and a geostatistical analysis of the 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.

2. Introduction

Ubiquity of spatially indexed datasets in various disciplines (Gelfand et al., 2010; Cressie and Wikle, 2015; Banerjee et al., 2014) has motivated researchers to develop variety of methods and models in spatial statistics. In most of the spatial applications, the interest lies in producing an estimate of the mean function and uncertainty intervals across the entire

area. In spatial data pertaining to many such applications, one often observes features at a global scale accompanied with local level variations. For example, in modeling the sea surface temperature data in the Eastern Pacific one must take into account large scale features such as the lower temperature in the shore of Canada than in the west coast of USA and also a number of local variations such as the small local variation in temperature due to upwelling along the California coast. It is desirable that models built upon such data enable capturing both large scale spatial variations and features at the local scale. Gaussian processes offer a rich modeling framework and are being widely deployed 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 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 and refer to Heaton et al. (2018) for a more comprehensive review. There are, broadly speaking, two different premises for modeling large spatial datasets. One of them is "sparsity", while the other is "dimension-reduction". Sparse methods include covariance tapering (see, e.g., Furrer et al. (2006); Kaufman et al. (2008); Du et al. (2009); Shaby and Ruppert (2012)), which introduces sparsity in the Gaussian covariance matrix using compactly supported covariance functions. This is effective for fast parameter estimation and interpolation of the response, but is less suited for more general inference on residual or latent processes due to exorbitantly expensive determinant computation of the sparse covariance matrix. An alternative approach introduces sparsity in the inverse of covariance (precision) matrix using conditional

independence assumptions or composite likelihoods (e.g., Vecchia (1988); Rue et al. (2009); Stein et al. (2004); Eidvisk et al. (2014); Datta et al. (2016); Guinness (2016)). In related literature pertaining to computer experiments, localized approximations of Gaussian process models are proposed, see e.g. Gramacy and Apley (2015). This literature is less model based and has different goals compared to the spatial literature.

Dimension-reduction methods subsume the popular "low-rank" models which express the realizations of the 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 for 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 so that 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). There are variants of dimension-reduction methods that partition the large spatial data into subsets containing fewer observations, run Gaussian processes in different subsets in parallel followed by combining inference from subsets, see e.g. Guhaniyogi and Banerjee (2017); Guhaniyogi et al. (2018). These methods allow an approximation of a full Gaussian process to be fit to very large datasets. Another important aspect of spatial modeling is the treatment of non-stationary covariance functions which allows variability to change over space. This explicitly changing structure in space is a desirable feature in models for non-stationary processes.

Multiresolution process models are introduced in the literature which layer multiple processes, usually non-stationary, on top of each other at different resolutions, with the idea

that higher resolution models can capture small scale behavior, while the lower resolutions capturing large scale behavior. There are some approaches which aim at modeling the spatial surface at multiple scales (Liang et al., 2008; Banerjee and Finley, 2007), however, literature on Bayesian multiscale spatial models for big data is quite insufficient.

Our approach combines the representation of a random field using compactly supported multiresolution basis functions with basis coefficients modeled using a newly developed *multiscale tree shrinkage prior*. The multiscale tree shrinkage prior is equipped to impart increasing shrinkage on basis coefficients as resolution increases. This effectively leads to a continuous analogue to selecting the number of resolutions necessary for modeling a subdomain. The proposed framework 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 only exists in part of the field, and if the field is nonstationary. Compactly supported basis functions and the computational strategy described in Section 4.1 evoke fast Bayesian estimation that only involves inverting a large number of small matrices in parallel. The proposition of the tree shrinkage prior that effectively shrinks a class of parameters having an inherent tree structure is novel in its own right with possible applications anticipated in statistical genomics and neuroscience, for example identifying main effects versus interaction effects in genetic studies.

It is noteworthy that a few other important articles on multiscale spatial models for big data have already appeared in the literature, see e.g. Nychka et al. (2015); Katzfuss (2017) and references therein. Although our approach has some similarities with the recently developed `LatticeKrig` model (Nychka et al., 2015) there are some important differences between these two classes of models. While `LatticeKrig` constrains the total contribution

to the variance from basis functions corresponding to the rth resolution to be of the order of r^{-v}, we propose to use the novel shrinkage prior distribution on basis coefficients to achieve similar goals. Both Nychka et al. (2015); Katzfuss (2017) allow for nonstationary covariance functions, but enforce the same multiresolution structure across the entire field. In contrast, our framework allows differential shrinkage of basis coefficients in different subdomains. Secondly, unlike `LatticeKrig`, the proposed multiscale approach incorporates data dependent choice of the kernel width. Thirdly, the proposed multiscale model can be naturally embedded in a hierarchical structure to model non-Gaussian data, as we detail out in the main article and in the web appendix. To the best of our knowledge, *LatticeKrig* with non Gaussian data is largely unexplored. Bayesian implementation of our approach naturally leads to effective characterization of uncertainty. Finally, due to the simple structure of our model, we have been able to show both large support property and posterior consistency for the proposed approach. These are desirable theoretical properties largely unexplored in most of the competing multiresolution models.

The remainder of the manuscript evolves as follows. Section 3 outlines the multiscale kernel convolution model development including the choice of knots, basis functions, basis coefficients and priors on them. Section 4 discusses posterior computation strategies and computation complexities. Detailed simulation studies with both Gaussian and non Gaussian data are shown in Section 5. Section 6 details out analysis of a massive sea surface temperature data in pacific ocean. Finally, Section 7 discusses what the newly developed multiscale model achieves, and proposes a number of future directions to explore. Theoretical insights and extensions to binary regression model are offered in the Web Appendix.

3. Multiscale Spatial Kriging

3.1 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 $\mathcal{D} \subseteq \mathbb{R}^d$, $d \in \mathbb{N}^+$. This article focuses on $d = 1, 2$. We assume the spatial process $w(\boldsymbol{s})$ follows a Gaussian process. One may construct a Gaussian process $w(\boldsymbol{s})$ over $\mathcal{D}$ by convolving a continuous white noise process $u(\boldsymbol{s})$, $\boldsymbol{s} \in \mathcal{D}$ with a smoothing kernel $K(\boldsymbol{s}, \phi)$ (ϕ might be space varying) so 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 $K(\cdot)$. A discrete approximation of it is obtained by sampling the convolved processes on a grid. Letting $\boldsymbol{s}_1^*, ..., \boldsymbol{s}_J^*$ be a set of knots in $\mathcal{D}$, a discrete approximation of $w(\boldsymbol{s})$ is given by

$$\theta(\boldsymbol{s}) = \sum_{j=1}^{J} K(\boldsymbol{s} - \boldsymbol{s}_j^*, \phi)u_j, \tag{3.1}$$

where u_j's are basis coefficients. The J knots are typically placed in a grid in $\mathcal{D}$, though other placements of knots have appeared in the literature. Varying the choice of the kernel functions and coefficients u_j, a rich variety of processes emerge from (3.1). Following Lemos and Sanso (2009), we term (3.1) as Discrete Convolutions of Terms (DCT). When J is small DCT provides computationally convenient approximation of the Gaussian process $w(\boldsymbol{s})$. However, Smaller J would greatly reduce approximation accuracy, while moderately large J exacerbates computational burden. The computational challenges cannot be solved by brute-force use of high-performance computing systems, and approximations or simplifying assumptions are necessary. One compelling idea to both reduce computation and increase approximation accuracy may come from using DCT at multiple scales. Next few sections carefully develop a multiscale-DCT model.

3.2 Partition of domain and choice of knots

To define the multiscale-DCT, we partition $\mathcal{D}$ into mutually exclusive and exhaustive sub-domains in resolution 1. In resolution 2, each of these sub-domains are partitioned into mutually exclusive and exhaustive sub-domains, and this process continues up to resolution R. At the lowest level, one partitions $\mathcal{D}$ into $J(1)$ subsets $\mathcal{D}_1, ..., \mathcal{D}_{J(1)}$. In the second level, each $\mathcal{D}_i$ undergoes P partitions so that the total number of partitions in the second level is $PJ(1)$. Likewise, let in the $(r-1)$th level the set of partitions can be described as $\{\mathcal{D}_{i_1,..,i_{r-1}} : i_1 \in \{1, 2, .., J(1)\}, i_2, ..., i_{r-1} \in \{1, ..., P\}\}$. In the rth level each $\mathcal{D}_{i_1,..,i_{r-1}}$ is partitioned into P subsets $\mathcal{D}_{i_1,..,i_{r-1},1}$,..,$\mathcal{D}_{i_1,..,i_{r-1},P}$, so that $\mathcal{D}_{i_1,..,i_{r-1}} = \bigcup_{i_r=1}^{P} \mathcal{D}_{i_1,..,i_{r-1},i_r}$, $\mathcal{D}_{i_1,..,i_{r-1},s} \bigcap \mathcal{D}_{i_1,..,i_{r-1},s'} = \phi, \ \forall\, s \neq s'$. Therefore, the number of partitions at the rth resolution is $J(r) = P^{r-1}J(1)$. In one dimensional ($d = 1$) case $\mathcal{D}_{i_1,..,i_{r-1},i_r}$'s are typically intervals, and bisection method is adopted to partition each interval into equal sized subintervals for the next resolution, i.e. $P = 2$. 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 resolution is typically a rectangle (though other choices are also possible), and each of them is divided into 4 equal sized subsets, i.e. $P = 4$ and $J(r) = 4^{r-1}J(1)$. This is a common practice to divide the domain into sub-domains as is observed in earlier multi-scale modeling literature, see e.g. Katzfuss (2017). Partitioning of the domain can be envisioned as formation of a tree, with sub-domains $\mathcal{D}_{i_1,...,i_r}$'s as nodes of the tree. Lower and higher resolutions correspond to the upper and lower nodes of this tree. $\mathcal{D}_1$,...,$\mathcal{D}_{J(1)}$ correspond to uppermost nodes of the tree. P branches emerge from each of these nodes leading to P^2 nodes in the second level of the tree and this process continues. Indeed, for any $i_1, ..., i_r$, $1 \leq r \leq R$, we define

$Subtree(\mathcal{D}_{i_1,..,i_r})$ by

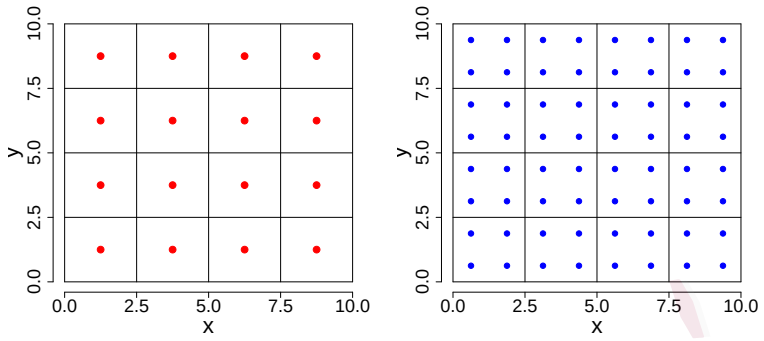
$$Subtree(\mathcal{D}_{i_1,..,i_r}) = \{\mathcal{D}_{i_1,..,i_r}\} \cup_{j=1}^{R-r-1} \{\mathcal{D}_{i_1,..,i_r,i_{r+1},..,i_{r+j}} : i_{r+1}, ..., i_{r+j} \in \{1, ..., P\}\}$$
$$\cup \{\mathcal{D}_{i_1,..,i_R}\}. \tag{3.2}$$

$Subtree(\mathcal{D}_{i_1,..,i_r})$ consists of all sub-domains of $\mathcal{D}_{i_1,..,i_r}$ in higher than rth resolution, including itself. Evidently, $Subtree(\mathcal{D}_{i_1,..,i_R}) = \mathcal{D}_{i_1,..,i_R}$. On a similar note, we also define the *father* node of $\mathcal{D}_{i_1,..,i_r}$ as the node $\mathcal{D}_{i_1,..,i_{r-1}}$.

Defining multiscale-DCT also requires choosing a set of knot points at every level. The knots $\boldsymbol{s}_1^1, ..., \boldsymbol{s}_{J(1)}^1$ in the first level is placed at the centers of $\mathcal{D}_1, ..., \mathcal{D}_{J(1)}$. Likewise, knots $\boldsymbol{s}_1^r, ..., \boldsymbol{s}_{J(r)}^r$ are kept at the centers of the partitions at the rth level. Technically knots can be placed at any point in the sub-domains. However, setting up parallel computation for the proposed model becomes easier when knots are put at the centers. To elaborate on it further, we refer to section 4.1 where the computational complexity of the method using parallelization is detailed out.

There is a one to one correspondence between the set of knots and the set of partitions of $\mathcal{D}$. Henceforth, we will interchangeably use *Subtree* and *Father* of a sub-domain with *Subtree* and *Father* of the knot that resides at the midpoint of that sub-domain, e.g. if $\boldsymbol{s}_j^r \in \mathcal{D}_{i_1,...,i_r}$, then $Subtree(\boldsymbol{s}_j^r)$ and $Father(\boldsymbol{s}_j^r)$ are synonymous with $Subtree(\mathcal{D}_{i_1,...,i_r})$ and $Father(\mathcal{D}_{i_1,...,i_r})$ 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 $\boldsymbol{s}_j^r$, $j = 1, .., J(r)$, belongs to $\mathcal{D}_{i_1,...,i_r}$ if $j = \sum_{l=1}^{r-1}(i_l - 1)P^{r-l} + i_r$. With this notation, $\boldsymbol{s}_k^{r-1}$ is the father node of $\boldsymbol{s}_j^r$ iff $k = \sum_{l=1}^{r-2}(i_l - 1)P^{r-l} + i_{r-1}$, i.e. $k = \lfloor \frac{j-1}{P} \rfloor + 1$, where $\lfloor x \rfloor$ is the greatest integer less than x.

To give 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



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

(2d) (2d)

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

with knots placed in the center of each rectangle. The number of knots in the first resolution is $J(1) = h_x \times h_y$. In resolution 2, every rectangle in the first resolution is divided into four congruent rectangles. Knots in the second resolution are placed at the centers of these new rectangles. Clearly, the distance between two horizontally and two vertically adjacent knots are $(h_2 - h_1)/h_x$ and $(h_4 - h_3)/h_y$ respectively, in 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 $R = 2$, $h_x = 4, h_y = 4, h_1 = 0, h_3 = 0, h_2 = 10, h_4 = 10$, $J(1) = 16$ for two dimensional surfaces. Hereon, we fix the template of domain partitions and the placement and number of knots in each partition.

3.3 Multiscale spatial process with radial basis functions

We model the spatial effects by a multi-scale DCT with R resolutions, rth resolution being modeled by a DCT with kernel $K(\cdot,\cdot,\phi_r)$, knots $\boldsymbol{s}_1^r,...,\boldsymbol{s}_{J(r)}^r$ and coefficients $\beta_1^r,...,\beta_{J(r)}^r$, $r = 1,..,R$. To elaborate on it further, the spatial surface $w(\boldsymbol{s})$ is written as $w(\boldsymbol{s}) = \sum_{r=1}^{R} w_r(\boldsymbol{s})$,

$$w_r(\boldsymbol{s}) = \sum_{j=1}^{J(r)} K(\boldsymbol{s}, \boldsymbol{s}_j^r, \phi_r)\beta_j^r. \tag{3.3}$$

ϕ_r represents the scale parameter for the rth resolution. The choice of ϕ_r's are detailed out later. The multiscale model represents the spatial effect with basis functions at multiple scales. The basis functions at lower resolutions have larger range parameters ϕ_r so that lower resolutions capture variability at large distances. On the other hand, the basis functions corresponding to higher resolutions have smaller range parameters ϕ_r so that finer local level variabilities are captured by the basis functions at higher resolutions. We formally discuss the choice of the basis functions and corresponding range parameters ϕ_r as below.

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

$$K(\boldsymbol{s}, \boldsymbol{s}_j^r, \phi_r) = \kappa\left(\frac{||\boldsymbol{s}-\boldsymbol{s}_j^r||}{\phi_r}\right) = \left(1 - \frac{||\boldsymbol{s}-\boldsymbol{s}_j^r||}{\phi_r}\right)_+^{l+1}\left[1 + (l+1)\frac{||\boldsymbol{s}-\boldsymbol{s}_j^r||}{\phi_r}\right]. \tag{3.4}$$

Geometrically, the kernel function consists of bumps centered at the node points with interpolation of the spatial surface at $\boldsymbol{s}$ in the rth resolution is 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 4.1 describes computational advantages derived from the compact support of this kernel.

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

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* r*th 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* $f(\cdot)$. $L_2(\mathcal{R}^d), C(\mathcal{R}^d)$ *correspond to set of all square integrable and set of all continuous functions respectively.*

Remark: Roughly speaking, the 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. Since the kernels in lower resolutions are meant to capture long range variabilities, one naturally imposes the constraint $\phi_1 > \phi_2 > \cdots > \phi_R > 0$. Secondly, given β_j^r, $j = 1, ..., J(r)$, $r = 1, ..., R$, ϕ_r determines the set of knots in the neighborhood of $\boldsymbol{s}$ for interpolating the spatial surface at $\boldsymbol{s}$. One could possibly keep ϕ_r as a parameter and update them using MCMC. Our detailed investigation reveals that such an approach adds unnecessary computational burden with no substantial inferential advantage. Therefore, this article fixes

$\phi_r = \eta||\boldsymbol{s}_j^r - \boldsymbol{s}_{j-1}^r||$, $\eta > 0$. η is a tuning parameter that determines the computational advantage vis a vis long range spatial dependence between observations. Since η is not a parameter of interest, the present article does not attempt to make full scale Bayesian inference on η. Rather at each step of the MCMC iteration, posterior likelihood is maximized over a grid of η. We further elaborate on it in Section 4.1.

3.4 Mutiscale spatial regression model

Our proposed mutiscale 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

$$y(\boldsymbol{s}) = \boldsymbol{x}(\boldsymbol{s})'\boldsymbol{\gamma} + \sum_{r=1}^{R}\sum_{j=1}^{J(r)} K(\boldsymbol{s}, \boldsymbol{s}_j^r, \phi_r)\beta_j^r + \epsilon(\boldsymbol{s}), \;\; \epsilon(\boldsymbol{s}) \sim N(0, \sigma^2), \tag{3.5}$$

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

3.5 Mutiscale shrinkage prior on β_j^r

Once the model formulation is complete, attention turns to assigning prior distributions on $\beta_j^r, \boldsymbol{\gamma}, \sigma^2$. While the prior specification on $\boldsymbol{\gamma}$ and σ^2 is straightforward, specifically $\boldsymbol{\gamma}$ is assigned a noninformative prior and $\sigma^2 \sim IG(c, d)$, constructing 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 practically do not require higher

resolutions for modeling. Mathematically, this amounts to setting $\beta_j^r = 0$ corresponding to the knots s_j^r located in the latter regions. It is also natural to assume that if rth resolution deemed unnecessary to model the surface in a sub-domain, any lth resolution for $l > r$ should be unnecessary too to model 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's finds equivalence in the variable selection literature in high dimensional regression. The goal in variable selection literature lies in identifying predictors not related to the response, equivalently the predictors having zero coefficients. A rich variety of methods have been proposed ranging from penalized optimization methods, such as Lasso (Tibshirani, 1996) and elastic net (Zou and Hastie, 2005), to Bayesian variable selection or shrinkage methods. The Bayesian approach is attractive in its probabilistic characterization of uncertainty for regression coefficients in high dimensions and for the resulting predictions.

An impressive variety of Bayesian shrinkage priors for ordinary high dimensional regressions with scalar/vector response on high dimensional vector predictors has been proposed in recent times, see for example Armagan et al. (2013); Hans (2009); Park and Casella (2008); Polson and Scott (2012); Carvalho et al. (2009) and references therein. The most popular and scalable class of high dimensional shrinkage priors does not set predictor coefficients zero a-posteriori. Rather these shrinkage priors are based on the principle of artfully 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 the prior to impose more shrinkage on coefficients in higher resolutions a-priori. However, the literature on shrinkage priors that impose increasing shrinkage along resolutions is quite insufficient. This article introduces a *multiscale tree shrinkage prior* to achieve this objective. It proposes

$$\beta_j^r \sim N(0, \alpha_j^r);\ \alpha_j^1 = \delta_1^{-1}, \alpha_j^2 = \delta_1^{-1}\delta_{j,2}^{-1}, \alpha_j^r = \alpha_{\lfloor \frac{j-1}{P} \rfloor + 1}^{r-1}\delta_{j,r}^{-1};$$

$$\delta_1 \sim Gamma(2,1), \delta_{j,r} \sim Gamma(c,1), c > 2. \tag{3.6}$$

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

4. Posterior computation and inference

This section describes posterior computation and inference for multiscale DCT. The main task for inference remains that of obtaining the posterior distribution of the unknown coefficients β_j^r and $\delta_{j,r}$ $j = 1, ..., J(r)$ and $r = 1, ..., R$, $\boldsymbol{\gamma}$ and σ^2. The formulation of multiscale DCT is simple, so that all the parameters allow simple Gibbs sampling updates. Due to the space constraint, we present the posterior computation with full conditional distributions of the parameters in the web appendix. Once posterior distributions of the parameters are available, they are employed to estimate interpolation of the residual surface and perform spatial predictions. By crucially exploiting the conditional independence among several parameters and multi-resolution structure of the problem, we obtain inference with excellent time and memory complexity (Section 4.1 and 5), can take full advantage of distributed-

memory systems with a large number of nodes (Section 4.1), and is thus scalable to large spatial datasets.

4.1 Distributed computation, surface interpolation and prediction

An important advantage of the multiscale DCT is that it facilitates distributed computation with little communication overhead at a large number of nodes, each only dealing with a small subset of the data. To begin with, Section 1 in the web appendix indicates that posterior updating of $\boldsymbol{\gamma}, \sigma^2, \delta_{j,r}, \delta_1$ can be carried out rapidly without having to store the entire data in one processor. The main computational difficulty comes from updating of $\boldsymbol{\beta}$. Single updating of β_j^r introduces too much autocorrelation, while joint updating of $\boldsymbol{\beta}$ requires inverting $(\sum_{r=1}^R J(r)) \times (\sum_{r=1}^R J(r))$ 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 $m = 1, ..., J(1)$, define the *neighborhood function* $\mathcal{N}(m)$ of m by $\mathcal{N}(m) = \left\{j : ||\boldsymbol{s}_j^1 - \boldsymbol{s}_m^1|| < 2\eta\right\}$. Similarly, the *neighborhood data function* is defined as $\mathcal{N}_D(m) = \left\{j : ||\boldsymbol{s}_j^1 - \boldsymbol{s}_m^1|| < \eta\right\}$. Let $\boldsymbol{\beta}_{j,r}^{Subtree}$ be a vector composed of all elements in $\mathcal{B}_{j,r}^{Subtree}$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_{1,1}^{Subtree}, ..., \boldsymbol{\beta}_{J(1),1}^{Subtree})'$. Exploiting the fact that knots are placed at the midpoints of every subdomain at each resolution and that the basis functions are compactly supported, one obtains $\boldsymbol{\beta}_{m,1}^{Subtree}|- \stackrel{\mathcal{L}}{=} \boldsymbol{\beta}_{m,1}^{Subtree}|\boldsymbol{y}_{\mathcal{N}_D(m)}, \boldsymbol{\beta}_{\mathcal{N}(m),1}^{Subtree}, m = 1, ..., J(1)$.

Algorithm 1 describes details of the computation strategy we adopt. As per Algorithm 1, the computation involves $J(1)$ nodes with 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 involved in the mth node is in computing Cholesky decomposition of a $dim(\boldsymbol{\beta}_{\mathcal{N}(m),1}^{Subtree}) \times dim(\boldsymbol{\beta}_{\mathcal{N}(m),1}^{Subtree})$ and multiplying a $dim(\mathcal{N}_D(m)) \times (\sum_{r=1}^R J(r))$ matrix with a vector of dimension $(\sum_{r=1}^R J(r))$. They incur

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 $(\boldsymbol{\beta}_{m,1}^{Subtree})^{(t)}$, $m = 1, ..., J(1)$, $\sigma^{2(t)}, \boldsymbol{\gamma}^{(t)}$, $\delta_{j,r}^{(t)}$, $j = 2, ..., J(r); r = 1, ..., R$ and $\delta_1^{(t)}$.

d. Maximize posterior likelihood w.r.t. $\eta \in \{1, ..., h_\eta\}$. 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 $\boldsymbol{K}_m = (K(\boldsymbol{s}, \boldsymbol{s}_1^1, \phi_1), ..., K(\boldsymbol{s}, \boldsymbol{s}_{J(R)}^R, \phi_R))$, $\boldsymbol{s} \in \mathcal{D}_m$.

g. Use the fact that $\sum_{m=1}^{J(1)} \boldsymbol{X}_m' \boldsymbol{X}_m = \boldsymbol{X}'\boldsymbol{X}$ and $\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\beta} = (\boldsymbol{y}_1 - \boldsymbol{K}_1\boldsymbol{\beta}, ..., \boldsymbol{y}_{J(1)} - \boldsymbol{K}_{J(1)}\boldsymbol{\beta})'$ 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.

computation complexities of $O(dim(\mathcal{N}(m))^3)$ and $O(dim(\mathcal{N}_D(m)) \sum_{r=1}^R J(r))$ respectively. Since $dim(\boldsymbol{\beta}_{\mathcal{N}(m),1}^{Subtree}) = ((2d)^R - 1)/(2d - 1)$, the computation time for the former is low. Choosing $J(1)$ large enough one 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

$$p(\boldsymbol{y}(\boldsymbol{s}_0) \,|\, \boldsymbol{y}) = \int p(\boldsymbol{y}(\boldsymbol{s}_0) \,|\, \boldsymbol{y}, \boldsymbol{\Theta}) p(\boldsymbol{\Theta} \,|\, \boldsymbol{y}) \, d\boldsymbol{\Theta}, \tag{4.7}$$

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 $\{\boldsymbol{\Theta}^{(t)}\}$, $t = 1, 2, \ldots, L$, 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 $\boldsymbol{y}(\boldsymbol{s}_0)^{(t)}$, $t = 1, 2, \ldots, L$ are samples from (4.7). This is

especially simple for multiscale DCT as $p(\boldsymbol{y}(\boldsymbol{s}_0) \mid \boldsymbol{\Theta})$ turns out to be a normal distribution.

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

5. Simulation studies

This section uses synthetic datasets to assess model performance with regard to interpolating unobserved residual spatial surface and predicting at new locations. To begin with, we present a one dimensional simulation experiment on a large dataset. The one dimensional simulation experiment helps to build intuition on how different resolutions capture large and small scale variabilities, including the advantage of choosing the tree shrinkage prior. Subsequently, we present a two dimensional example where computation time and performance of multiscale DCT (MDCT) will be compared with state-of-the-art and popular spatial models for big data. A non-distributed implementation of the methods are carried out in `R` version 3.3.1 on a 16-core machine (Intel Xeon 2.90GHz) with 64GB RAM.

5.1 One dimensional Example

For the one dimensional example, we simulated a dataset 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

$$w_0(s) = \begin{cases} \sin(2\pi s)s, \text{ if } 0 \leq s < 2 \\ |\sin(s-3)|^3, \text{ if } 2 \leq s < 4 \\ 5|s-5|, \text{ if } 4 \leq s < 6 \\ \sin(2\pi s)s, \text{ if } 6 \leq s < 10. \end{cases} \tag{5.8}$$

Here $s_1, ..., s_n$ are the set of spatial locations, $\boldsymbol{w}_0 = (w_0(s_1), ..., w_0(s_n))'$, $\boldsymbol{\gamma} = (\gamma_0, \gamma_1)$, $\boldsymbol{y} = (y(s_1), ..., y(s_n))'$ and $\boldsymbol{X} = (\boldsymbol{x}(s_1)' : ... : \boldsymbol{x}(s_n)']'$, $\boldsymbol{x}(s_i) = (1, x(s_i))$. $x(s_i)$'s are drawn 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.

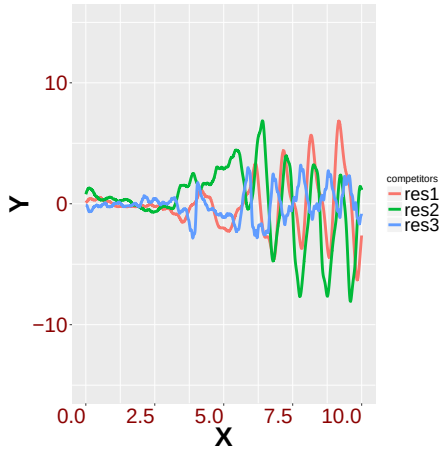
We fit MDCT with $J(1) = 30$ to this dataset. As competitors we implement

DCT-GDP: DCT-GDP uses the same basis functions as multiscale DCT, but replaces *multiscale tree shrinkage prior* by Generalized Double Pareto (GDP, Armagan et al. (2013)) shrinkage prior on the basis coefficients.

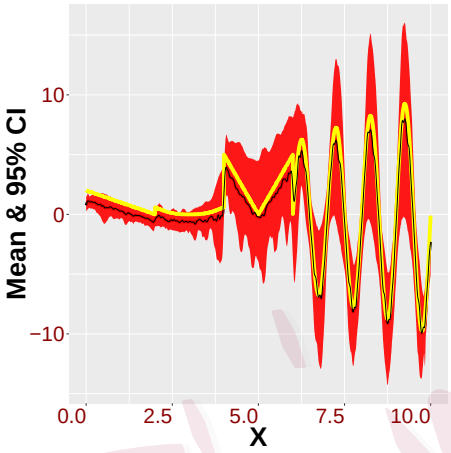
DCT-Normal: DCT-Normal also uses the same basis functions with the prior on basis coefficients given by the independent normal distributions.

DCT-GDP and DCT-normal are mainly aimed at comparing the inferential advantage of the tree shrinkage prior over the ordinary shrinkage prior and normal prior distributions respectively. Additionally, we fit multi-scale DCT with one and two resolutions to assess how the choice of $R = 3$ improves inference. Multiscale DCT with one and two resolutions are referred to as MDCT(1) and MDCT(2) respectively.

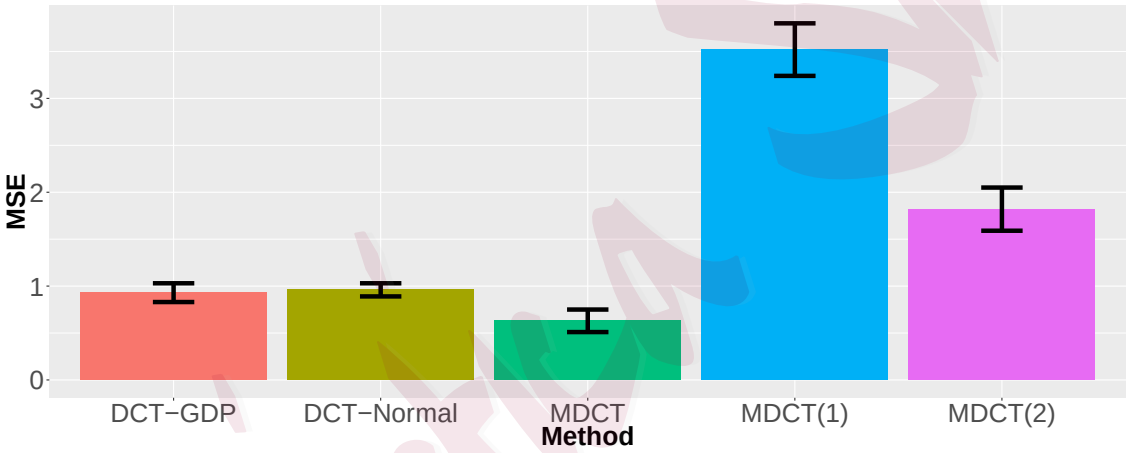
Figure 2 reveals the role played by three resolutions in estimating $w_0(s)$. While resolution



(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 function in $R = 3$ resolutions. The true function is in yellow and the estimated function is in black. 95% confidence bands for the estimated function are displayed in red. (c) shows the MSE with associated standard errors for all competitors.

1 mostly captures positive side of the sinusoidal curve, negative extremities of the sinusoidal curve is mostly reconstructed by resolution 2. Resolution 3 captures the local variability in the interval [4,10].

The inferential performance of MDCT is evaluated in estimating the spatial surface using mean squared error. To be more precise, let $\hat{w}(s_i)$ be the posterior median of $w(s_i)$. Define mean squared error (MSE) by $MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{w}(s_i) - w_0(s_i))^2$. Average MSE along with the associated standard errors over multiple simulations for the three competing models are presented in Figure 2. It is evident from these figures that MDCT, with the same number of knots and same basis functions, provide improved inference, the reason being implementation of a structured prior distribution on the basis coefficients. The computation times to implement the three competitors are about the same, with one MCMC iteration in MDCT taking ≈ 0.33 seconds to run the full scale inference. Additionally, there seems to be a substantial improvement in terms of MSE with increasing resolutions, though it stabilizes after $R = 3$.

The one dimensional exploration of MDCT shows that multi-scaling is able to capture local features succinctly, yielding superior inference with similar number of knots and same basis functions over one scale DCT. Also, the computational advantage of multiscale DCT is enormous given that full Bayesian inference can be performed with a series of local computations. Moreover, the architecture of MDCT allows storing subsets of data in different processors. In the next section a more involved comparative analysis of MDCT with popular competitors is presented in the context of two dimensional spatial examples.

5.2 Two dimensional example

5.2.1 Two dimensional example with Gaussian data

This section uses two dimensional synthetic datasets to assess the performance of MDCT in comparison to popular models for large spatial data. For the sake of our exposition, MDCT

is implemented with 3 resolutions having a total of 2100 basis functions. As competitors to MDCT we implement:

(1) **Modified predictive process (MPP)**: MPP (Finley et al., 2009; Banerjee et al., 2010) is a low rank method implemented using the package `spBayes` in `R`.

(2) **LatticeKrig**: `LatticeKrig` package in `R` is employed for non-Bayesian implementation of LatticeKrig (Nychka et al., 2015) with 3 resolutions having a total 12678 basis functions.

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

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

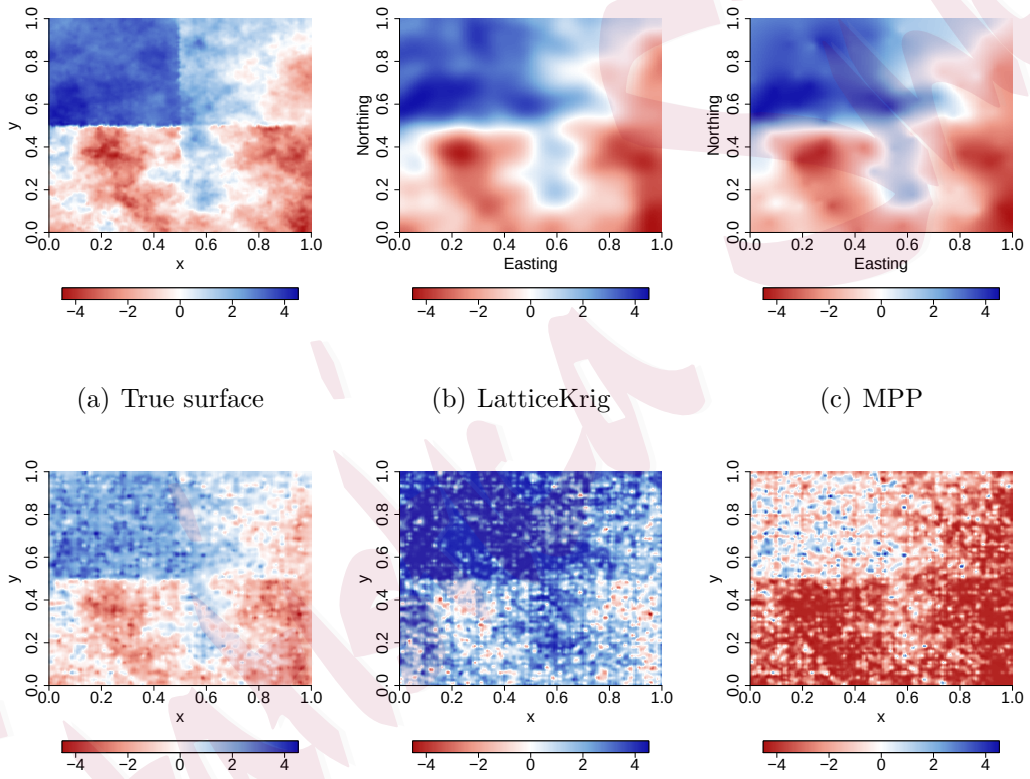
$$y(\boldsymbol{s}) = \boldsymbol{x}(\boldsymbol{s})'\boldsymbol{\gamma} + w_1(\boldsymbol{s})I(s_1 < 0.5, s_2 < 0.5) + w_2(\boldsymbol{s})I(s_1 < 0.5, s_2 > 0.5) + w_3(\boldsymbol{s})I(s_1 > 0.5, s_2 < 0.5)$$
$$+ w_4(\boldsymbol{s})I(s_1 > 0.5, s_2 > 0.5) + \epsilon(\boldsymbol{s}), \;\; \epsilon(\boldsymbol{s})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

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, $\boldsymbol{\gamma} = (\gamma_0, \gamma_1)$. $w_j(\boldsymbol{s})$ $(j = 1, .., 4)$ follows Gaussian processes with mean 0 and covariance kernel $\upsilon(\boldsymbol{s}, \boldsymbol{s}', \theta_1, \theta_2, \nu))$ chosen from the popular Matern class of correlation functions given by

$$\upsilon(\boldsymbol{s}, \boldsymbol{s}', \theta_1, \theta_{2j}, \nu) = \frac{\theta_1}{2^{\nu-1}\Gamma(\nu)}(||s - s'||\theta_{2j})^{\nu}\mathcal{K}_{\nu}(||s - s'||\theta_{2j});\; \theta_{2j} > 0,\; \nu > 0, \tag{5.9}$$

with θ_{2j}, ν controlling spatial decay and process smoothness respectively, Γ is the usual

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 non-differentiable sample paths. Additionally, $w_j(\boldsymbol{s})$'s are assigned varying spatial decay parameters with $\theta_{21} = 1.5, \theta_{22} = 0.1, \theta_{23} = 1, \theta_{24} = 0.5$. Among $10,500$ observations, 10000 are randomly selected for model fitting and the rest are kept as a test dataset to facilitate predictive inference.



(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 along with posterior mean residual surface from (b) LatticeKrig (c) modified predictive process (d) multiscale DCT and; (e) and (f) present estimated 95% upper and lower quantile surfaces

Figure 3 presents true data generating surface and the estimated residual surfaces for

LatticeKrig (LK), MDCT and MPP. MPP shows little oversmoothing, while LatticeKrig and multiscale DCT yield essentially equivalent degree of precision in terms of the residual surface estimation. 95% credible interval for the residual surface of MDCT fits tightly around the median surface.

Next, we turn our attention to the predictive inference of the competitors. To this end, we compare all competitors based on their ability to produce accurate point prediction and predictive uncertainties. Point prediction of the competitors are judged based on the mean squared prediction error (MSPE) metric. For Bayesian competitors, i.e. MDCT and MPP, predictive uncertainties are characterized by the length and coverage of 95% predictive intervals. The frequentist implementation of LK provides predictive point estimates and standard errors, while LaGP yields posterior predictive mean and standard error (SE) at the unobserved locations. Thus, for these two competitors, approximate 95% predictive intervals are constructed by considering predictive point estimate $\overset{+}{-}\, 1.96\, SE$, which provides an approximate 95% predictive interval.

Figure 4 shows that MDCT yields essentially equivalent MSPE with those of LK and LaGP. Interestingly, MDCT with $R = 3$ resolution shows significantly improved performance (MSPE of 1.63) compared to MDCT with $R = 1$ (MSPE of 1.98) and $R = 2$ (MSPE of 1.85). Intuitively, one can explain such an upsurge in performance by noting that the true surface generated as a mixture of four region specific Gaussian processes with three out of four regions exhibiting significant local behavior. In terms of characterizing predictive uncertainties, MDCT exhibits marginal over-coverage with wider 95% credible interval compared to the other competitors. This is not surprising given the degree of complexity embedded in the MDCT model with a large number of parameters. MPP demonstrates little under-coverage

with a narrower predictive interval compared to the other competitors. LK and laGP also show competitive performance with close to the nominal coverage, though the 95% predictive interval from LK is only justifiable through normal approximation.

To check the sensitivity with respect to the choice of $R = 3$, we run our analysis with $R = 4, 5$ and compare to 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 this conclusion to hold across a number of simulation studies. Therefore $R = 3$ is kept throughout this article.

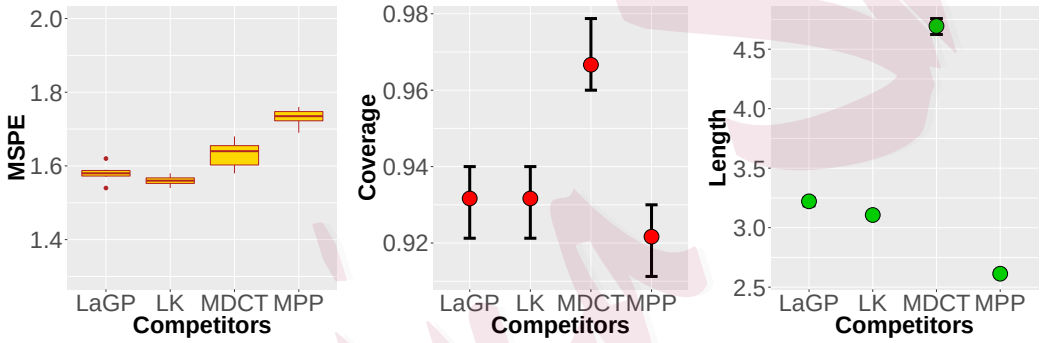


Figure 4: Plot at the top indicates boxplot of mean squared prediction error for all competitors over a few replications. Second and third plots show coverage and length of 95% predictive intervals for the competitors over the same replications.

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

Table 1: Average mean squared prediction error for MDCT with $R = 3, 4, 5$. Subscripts provide associated standard errors over 5 repeated simulations.

A few remarks are in order. Note that MDCT and LatticeKrig have 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 so as to yield similar point estimates with LatticeKrig with much less number of basis functions. Most importantly, while GMRF prior distributions may not very conducive to parallel computation, multiscale DCT is able to draw full scale Bayesian inference with a series of parallelizable local computations. It is important to mention that LaGP may not find an easy extension to non-Gaussian data or to hierarchical models, as it is not model based. Indeed, the performance of LK to non Gaussian data is yet unexplored, while MDCT can readily be embedded into a hierarchical structure to model non-Gaussian spatial data, or to blend different sources of information, as is described in the next section and in web appendix.

5.2.2 Two dimensional example with non Gaussian data

This section briefly describes performance of MDCT in presence of non Gaussian heavy tailed data distribution. For this specific simulation, $10,500$ locations $\boldsymbol{s}_1, ..., \boldsymbol{s}_n$ within $[0,1] \times [0,1]$ domain are drawn uniformly. 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 Students-t distribution with d.f. 2 and scaling parameter σ. $w_(\boldsymbol{s}), w_2(\boldsymbol{s})$ follow independent Gaussian processes with exponential covariance functions having range parameters 1.5 and 0.3 respectively. This simulations mimics the possible behavior of a random field that results from a variable that exhibits a distribution with long tails, over an area with a some sharp geographical boundaries, like a costline, a river or a mountain ridge. To this data, we fit MDCT with error following a Students-t error

distribution,

$$y_i = \boldsymbol{x}(\boldsymbol{s}_i)'\boldsymbol{\gamma} + \sum_{r=1}^{R}\sum_{j=1}^{J(r)} K(\boldsymbol{s}, \boldsymbol{s}_j^r, \phi_r)\beta_j^r + \epsilon_i, \;\; \epsilon_i \overset{iid}{\sim} ST_2(0, \sigma^2), \; i = 1, .., n. \tag{5.10}$$

Note that Students-t distribution assumes scale mixture of normal representation, which we exploit to set up efficient Gibbs sampler for MDCT. Details of the posterior computation is omitted due to space constraint and presented in the web appendix.

In absence of any open source implementation of LK and LaGP for non Gaussian data, we fit the ordinary LatticeKrig and laGP to assess the performance of MDCT. Figure 5 shows the estimated mean residual surface for MDCT along with LatticeKrig and the true surface. To our expectation, MDCT is able to identify local level spatial variation in the surface. Table 2

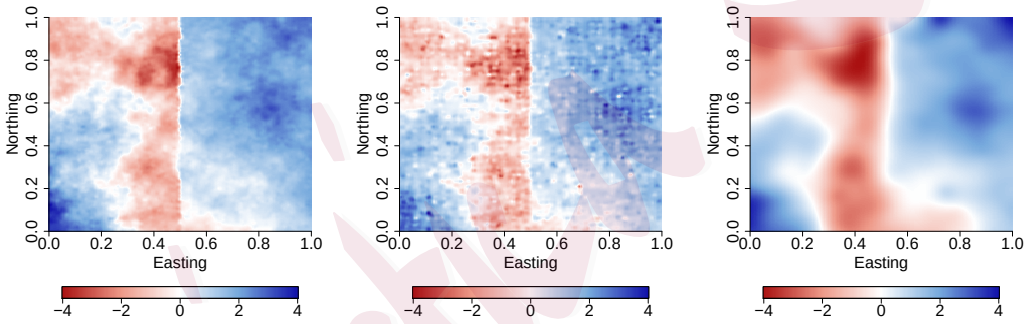


Figure 5: Left to right: True surface, median posterior surface of MDCT, median posterior surface of LatticeKrig.

displays MSPE, coverage and length of 95% predictive intervals for the three methods. laGP and LatticeKrig being fitted with the normal error assumption, demonstrates little under-coverage with narrower predictive intervals. MDCT shows wider predictive intervals with little over coverage. MDCT also performs little better than LK and laGP in terms of MSPE. The result perhaps indicate two important points. First, implementation of MDCT can be straightforward and accurate even with non Gaussian error. Second, and most importantly,

Bayesian implementation of MDCT under non Gaussian error is able to bring inferential advantages over its competitors. Implementation of MDCT with binary spatial model is given in the web appendix.

	MDCT	laGP	LatticeKrig
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

Table 2: Mean squared prediction error (MSPE), length and coverage of 95% predictive intervals of MDCT, laGP and LatticeKrig.

Computation Time: MDCT in this specific example takes approximately 3.07 seconds per iteration with non-optimized, non-parallel `R` implementation, while MPP implemented in `C++` takes close to 7.2 seconds to run one MCMC iteration. We notice, though, that MPP performs the estimation of the basis functions, while MDCT assumes a fixed form with the empirical Bayes estimate of η at every iteration. It is possible that, performing more elaborate inference on the kernel parameters of the MDCT would improve the predictive performance of the model. This would come at the cost of increased computational complexity, thus the benefits of such extension is unwarranted. In this specific example MDCT is implemented with $J(1) = 100$. To understand how the computation time of MDCT varies vis a vis MPP with changing n and $J(1)$, we implement both MPP and MDCT with $J(1) = 5^2, 10^2$ for different sample sizes. Figure 6 reports the computation time for the competitors using the `R` function `Sys.time`. It is be noted that MDCT can be implemented either by sequentially updating $J(1)$ blocks of parameters or by parallelly updating these $J(1)$ blocks independently

in $J(1)$ nodes. Thus the figure displays computation time with both parallel and non-parallel implementation of the MDCT model. Clearly, the computation time for MDCT increases linearly with n for both cases and MDCT with $J(1) = 5^2$ is about 4-5 times faster than $J(1) = 10^2$. The increase in computation time is due to sequential updating of parameters in $J(1)$ blocks. With a proper parallelized implementation of MDCT, arguably the increase in computation time from $J(1) = 25$ to $J(1) = 100$ will be minimal. We found that practical implementation of MPP becomes prohibitive due to both, memory allocation and exorbitant computation time, for n above $100,000$. It is worth noticing that non-Bayesian implementation of LK and LaGP draw inference for a point estimate within a few minutes. In summary, 2D simulation examples comprehensively establish MDCT as an effective tool for fast Bayesian implementation of large scale spatial data.

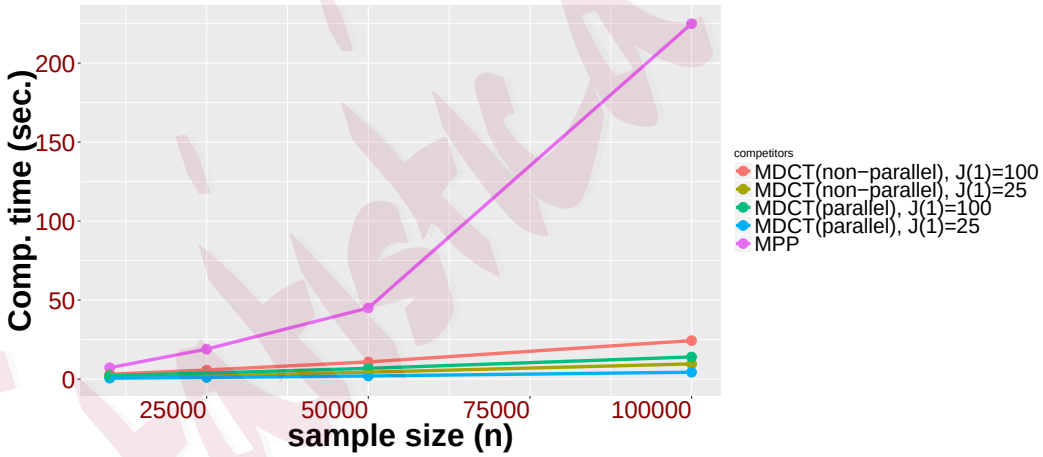


Figure 6: Computation time for MPP with 200 knots, MDCT with $J(1) = 25$ and $J(1) = 100$. Computation times per MCMC iteration are presented for both MDCT and MPP.

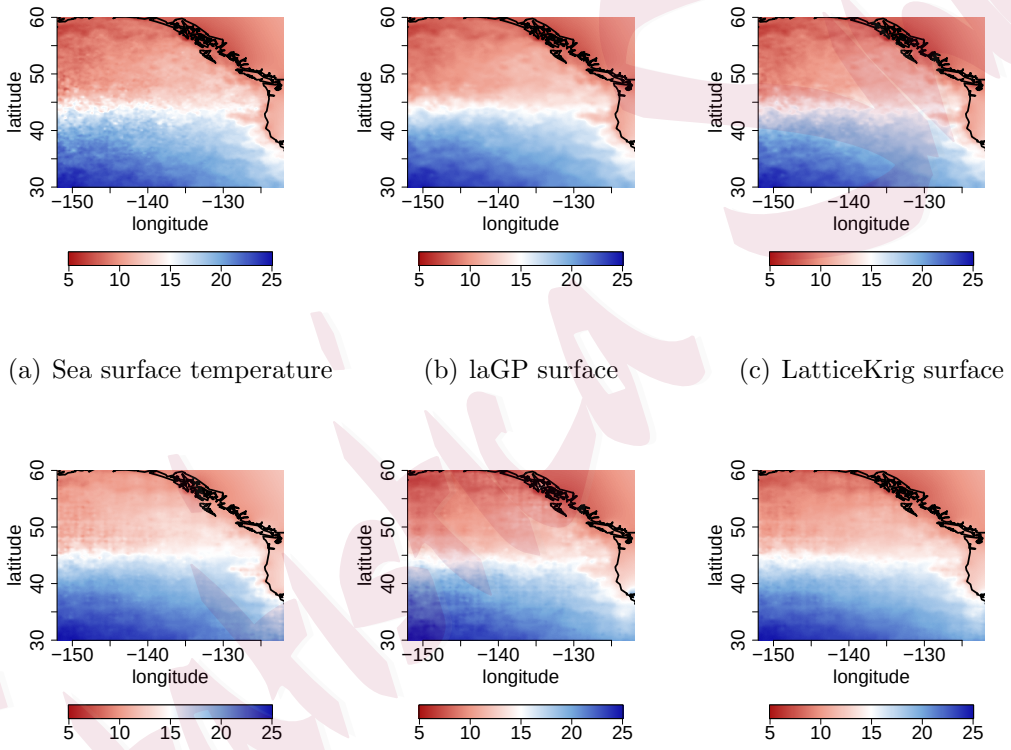
6. Analysis of sea surface temperature data

A description of 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 for the purpose of understanding the properties of water masses and their changes in time. They are also used to assess, initialize and constrain numerical models of the climate. Increasingly sophisticated climatological research requires, not only the 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 articles have appeared to address this issue in recent years, see e.g. Higdon (1998), Lemos and Sanso (2009), Lemos and Sanso (2006), Berliner et al. (2000), Wikle and Holan (2011).

This article considers the problem of capturing the spatial trend and characterizing the uncertainties in the sea surface temperature (SST) in the West coast of mainland USA, Canada and Alaska between $30^0 - 60^0$ N. latitude and $122^0 - 152^0$ W. longitude. The dataset is obtained from NODC World Ocean Database 2016 and we use the data collected in the month of October for all the spatial locations. Note that, for this example, we ignore the temporal component. We perform screening of the data to ensure quality control and then choose a random subset of $113,412$ spatial observations over the domain of interest. Out of the total observations, about 90%, i.e $100,000$ observations are used for model fitting and rest are used for prediction. We replicate this procedure 5 times to eliminate any chance factor in our analysis. The domain of interest is large enough to allow considerable spatial variation in SST from north to south and provides an important first step to extend these models for the analysis of global scale SST database.

The plot of the sea surface temperature along with coastal lines of Western United States

and Canada is shown in Figure 7(a). 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 non-spatial model with latitude and longitude as linear predictors using ordinary least square (OLS) and discover spatial dependence with no obvious pattern of aniosotropy. Thus a multiscale DCT model with latitude and longitude as predictors seem to be a desirable model for this data.



(a) Sea surface temperature (b) laGP surface (c) LatticeKrig surface

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

Figure 7: (a) Sea surface temperature in October 2016 for a portion of the North Pacific. Panels (b), (c) and (f) show the estimated mean predictive surfaces for three competing models. Figures (d) and (e) present the point-wise predictive bands for the MDCT. Temperatures are in degree centigrade.

The proposed MDCT model for the training data uses $R = 3$ resolutions with the first resolution having $J(1) = 100$ knots. To minimize edge effect, some knots are also kept inside land. We implement Algorithm 1 and run it for 2000 iterations to discover that $\eta = 1$ appears overwhelmingly among 2000 iterations. Thus to reduce unnecessary storage complexity and to speed up the computations for a dataset of this scale, we run the rest of the iterations with $\eta = 1$. The model thereafter is run for 5000 iterations, convergence diagnostics is performed with the `coda` package in R which indicates that 2000 iterations are sufficient as burn-in, to achieve practical convergence. As competitors to the MDCT, we fitted LatticeKrig and LaGP to the data. MPP is computationally prohibitive for the size of the dataset and is omitted from the comparison.

	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

Table 3: Mean squared prediction error (MSPE), length and coverage of 95% predictive intervals of MDCT, MDCT(1), MDCT(2), laGP and LatticeKrig.

The predictive power of the proposed model, along with that of its competitors, is assessed based on mean squared prediction error (MSPE), coverage and length of 95% predictive intervals. The non-spatial model and MDCT yield MSPE 1.34 and 0.18 respectively. The dramatic improvement in MSPE due to the inclusion of a spatial structure, that is evident from Table 3 corroborates the fact that 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 has a little better predictive performance than MDCT. The smallest MSPE in the table corresponds to LatticeKrig fitted with $R = 3$ resolutions. Overall, MDCT with $R = 3$ resolutions turns out to be a competitive performer in predictive inference. Importantly, even with non-parallel implementation MDCT takes about 26 seconds to run one iteration. As shown in Figure 6, the computation time can be reduced by multiple folds through efficient parallel implementation. On the contrary, even the frequentist implementation of LatticeKrig takes about 2 hours. It is evident that MDCT with $R = 3$ resolutions performs better than MDCT(1) and MDCT(2). However, fitting MDCT beyond $R = 3$ unnecessarily exacerbates computational burden with minimal improvement of inferential and predictive performance.

7. Conclusion and Future Work

This article proposes a novel multiscale kriging model for spatial datasets. 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 degree of smoothness. One key ingredient of our mutiscale model is the kernel convolution with a compactly supported kernel of minimal degree and knots placed in a regular grid at every resolution. Theoretically, it allows us to completely characterize the space of functions generated from the mutiscale spatial model. Another important contribution of the current article is to propose a new class of *multiscale tree shrinkage prior* distribution for the basis coefficients. The construction of a tree shrinkage prior is introduced with a consideration that as the model moves to the higher resolutions, more and more basis coefficients become irrelevant.

Besides the important methodological and theoretical contributions that the proposed

model entails, there is an equally important contribution in computational efficiency for large datasets. The research on multiscale spatial models is largely motivated by the quest to build a complex and flexible spatial model that allows accurate spatial inference and prediction for massive datasets and yet allows rapid Bayesian computation. The compactly supported kernel together with the multiscale shrinkage priors allow efficient MCMC of model parameters.

It is important to note that the current framework of mutiscale Bayesian modeling of spatial datasets can readily be extended to spatio-temporal datasets. Additionally, the recent idea of spatial meta kriging (Guhaniyogi and Banerjee, 2017) allows scalability by fitting a spatial model independently on partitions (disjoint subsets) of a big data followed by combining inferences from subsets. The proposed multiscale framework is able to scale up to $\approx$ half a million spatial locations, but may struggle with tens of million. If we have resources to run on $\approx H$ different subsets with each subset running our approach with n data points, then SMK combined with our approach can yield full Bayesian inference on $\approx nH$ locations. Finally, this article proposes one specific rectangular partition of the domain. There is a scope of future research as to how adaptive partitioning of the domain is to be implemented using techniques such as the Voronoi tesselation. Adaptive partitioning with the appropriate placement of knots might significantly reduce the number of knots required to yield acceptably accurate inference. We will explore these approaches in future.

Acknowledgements

The first author is partially supported by Office of Naval Research, award no. N00014-18-2741. The second authors 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).

8. Supplementary Material

Web Appendix includes

1. *Posterior computation for MDCT with Gaussian model.*
2. *Posterior computation for MDCT with non Gaussian data.*
3. *Two dimensional illustration of MDCT with binary spatial data.*
4. *Theoretical properties.*

References

Armagan, A., Dunson, D. B. and Lee, J. (2013). Generalized double pareto shrinkage. *Statistica Sinica 23*, pp. 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*, pp. 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*, pp. 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*, pp. 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*, pp. 3953–3968.

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

Cressie, N. and Johannesson, G. (2008). Fixed rank kriging for very large spatial data sets. *Journal of the American Statistical Association 70*, pp. 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*, pp. 800–812.

Du, J., Zhang, H. and Mandrekar, V. (2009). Fixed-domain asymptotic properties of tapered maximum likelihood estimators. *The Annals of Statistics 37*, pp. 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*, pp. 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*, pp. 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*, pp. 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*, pp. 561–578.

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*, pp. 1119–1130.

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*, pp. 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*, pp. 835–845.

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

Higdon, D. (2002). Space and space-time modeling using process convolutions. *Quantitative methods for current environmental issues, Springer*, pp. 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. 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*, pp. 201–214.

Kaufman, C. G., Schervish, M. J. and Nychka, D. W. (2008). Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association 103*, pp. 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*, pp. 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*, pp. 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*, pp. 579–599.

Park, T. and Casella, G. (2008). The Bayesian Lasso. *Journal of the American Statistical Association 103*, pp. 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*, pp. 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*, pp. 319–392.

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

Stein, M. L. (2007). Spatial variation of total column ozone on a global scale. *The Annals of Applied Statistics 1*, pp. 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*, pp. 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*, pp. 275–296.

Tibshirani, R. (2014). Regression shrinkage and selection via the lasso *Journal of the Royal Statistical Society Series B (Statistical Methodology) 58*, pp. 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*, pp. 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*, pp. 301–320.