Published in final edited form as:
J Comput Graph Stat. 2019 ; 28(2): 401–414. doi:10.1080/10618600.2018.1537924.

Efficient algorithms for Bayesian Nearest Neighbor Gaussian Processes

Andrew O. Finley[1], **Abhirup Datta**[2], **Bruce C. Cook**[3], **Douglas C. Morton**[3], **Hans E. Andersen**[4], **Sudipto Banerjee**[5]

[1]Michigan State University

[2]Johns Hopkins University

[3]National Aeronautics and Space Administration

[4]United States Forest Service

[5]University of California, Los Angeles

Abstract

We consider alternate formulations of recently proposed hierarchical Nearest Neighbor Gaussian Process (NNGP) models (Datta et al., 2016a) for improved convergence, faster computing time, and more robust and reproducible Bayesian inference. Algorithms are defined that improve CPU memory management and exploit existing high-performance numerical linear algebra libraries. Computational and inferential benefits are assessed for alternate NNGP specifications using simulated datasets and remotely sensed light detection and ranging (LiDAR) data collected over the US Forest Service Tanana Inventory Unit (TIU) in a remote portion of Interior Alaska. The resulting data product is the first statistically robust map of forest canopy for the TIU.

1 Introduction

As spatial statisticians confront massive datasets with locations ~10^6 and increasingly demanding inferential questions, several existing approaches that once seemed attractive for locations in the order of 10^4 become impractical. Recent methodological developments within the burgeoning literature on this subject aim to deliver massively scalable spatial processes. Sun et al. (2011) and Banerjee (2017) provide background and more current work (also see references therein), respectively, in this area. A recent contribution by Heaton et al. (2017) is particularly useful as it provides an overview of modeling approaches for large spatial data that are under active development, and a comparison of these approaches based on the analysis of a common dataset in the form of a “friendly competition.” In addition to Nearest Neighbor Gaussian Process (NNGP: Datta et al., 2016a) models, the comparison presented by Heaton et al. (2017) considered reduced rank predictive processes (Banerjee et al., 2008; Finley et al., 2009), covariance tapering (Furrer and Sain, 2010; Furrer, 2016), gapfilling (Gerber, 2017), metakriging (Guhaniyogi and Banerjee, 2018), spatial partitioning (Sang et al., 2011; Barbian and Assunção, 2017), fixed rank kriging (Cressie and Johannesson, 2008; Zammit-Mangion and Cressie, 2017), multiresolution approximation (Katzfuss, 2017), stochastic partial differential equations (Rue et al., 2017), lattice kriging (Nychka et al., 2015), and local approximate Gaussian processes (Gramacy and Apley, 2015; Gramacy, 2016). The comparison was based on out-of-sampled predictive performance and, to a lesser extent, computing time for a moderately sized simulated and real dataset comprising 105,569 observations. Comparisons showed NNGP models yielded highly competitive predictive performance and computation time.

With a few exceptions, e.g., Furrer and Sain (2010) and Gramacy (2016), the literature on scalable spatial process models has focused primarily on theoretical and methodological developments with little attention to the algorithmic details needed for effectively applying them. For example, Datta et al. (2016a) implement a "sequential" Gibbs sampler that involves updating a high-dimensional latent random effect vector and is prone to high autocorrelations and slow convergence. Most of the aforementioned articles do not discuss how researchers can, in practice, exploit high-performance computing libraries to obviate expensive numerical linear algebra (e.g., expensive matrix multiplications and factorizations) and deliver full Bayesian inference for massive spatial datasets. We address this gap for the NNGP models here by outlining three alternate formulations that are significantly more efficient for practical implementation than Datta et al. (2016a). Along with the accompanying code supplied with this manuscript, our intended contribution is well aligned with recent emphasis on reproducible research for challenging data analysis in the context of massive spatial datasets.

Our motivating scientific application concerns forest resource monitoring efforts and, in particular, to create fine resolution canopy height predictions using remotely sensed data collected at over 5 million locations. Spatially explicit estimates of forest canopy height are key inputs to a variety of ecosystem and Earth system modeling efforts (Finney, 2004; Hurtt et al., 2004; Stratton, 2006; Lefsky, 2010; Klein et al., 2015). These and similar applications seek inference about forest canopy height model parameters and predictions that can be propagated through subsequent computer models of ecosystem function to yield more robust error quantification. Bayesian inference is attractive here as it supplies full posterior predictive distributions for the outcomes and for the latent process at arbitrary locations in the region of interest.

The remainder of this article proceeds as follows. Section 2 provides a brief overview of NNGP models and their computational aspects. This is followed by three distinct and efficient alternate formulations: the collapsed NNGP model, a NNGP model for the outcomes themselves (with no latent process), and a conjugate NNGP model that allows MCMC-free inference. Section 3 offers detailed simulation experiments on model performance and assessment and also presents a detailed analysis of the US Forest Service Tanana Inventory Unit (TIU) dataset. Finally, Section 4 concludes the manuscript with a summary and an eye toward future work.

2 Nearest Neighbor Gaussian Processes

Let $y(\mathbf{s}_i)$ and $\mathbf{x}(\mathbf{s}_i)$ denote the response and the predictors observed at location $\mathbf{s}_i$, $i = 1, 2,\ldots, n$. A spatial linear mixed model posits $y(\mathbf{s}_i) = \mathbf{x}(s_i)^\top \boldsymbol{\beta} + w(\mathbf{s}_i) + \epsilon(\mathbf{s}_i)$, where the random effect $w(\mathbf{s}_i)$ sums up the effect of unknown or unobserved spatial covariates, and $\epsilon(\mathbf{s}_i)$ denotes the independent and identically observed noise. Gaussian Processes (GP) are commonly used for modeling the unknown surface $w(\mathbf{s})$. In particular, $w(\mathbf{s}) \sim GP(0, C(\cdot,\cdot|\boldsymbol{\theta}))$ implies that $\mathbf{w} = (w(\mathbf{s}_1), w(\mathbf{s}_2),\ldots, w(\mathbf{s}_n))^\top$ is Gaussian with mean zero and covariance $\mathbf{C} = (c_{ij})$, where $c_{ij} = C(\mathbf{s}_i, \mathbf{s}_j \mid \boldsymbol{\theta})$ and $\boldsymbol{\theta}$ denotes the GP covariance parameters. A popular choice for $C(\cdot, \cdot \mid \boldsymbol{\theta})$ is the Matérn covariance function specified as:

$$C\left(\mathbf{s}_i, \mathbf{s}_j; \sigma^2, \phi, \nu\right) = \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)}\left(\|\mathbf{s}_i - \mathbf{s}_j\|\phi\right)^\nu \mathscr{K}_\nu\left(\|\mathbf{s}_i - \mathbf{s}_j\|\phi\right); \quad \phi > 0, \nu > 0, \quad (1)$$

where $\boldsymbol{\theta} = \{\sigma^2, \phi, \nu\}$ and $\mathscr{K}$ denotes the Bessel function of second kind. Customary Bayesian hierarchical models are constructed as

$$p\left(\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\right) \times N(\mathbf{w}|\mathbf{0}, \mathbf{C}) \times N(\mathbf{y}|\mathbf{X}\boldsymbol{\beta} + \mathbf{w}, \tau^2\mathbf{I}), \quad (2)$$

where $p(\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2)$ is specified by assigning priors to $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ and τ^2. When n is very large, implementing (2) poses multiple computational roadblocks. Firstly, storing the matrix $\mathbf{C}$ requires $O(n^2)$ dynamic memory. Furthermore, evaluating $N(\mathbf{w} \mid \mathbf{0},\mathbf{C})$ involves factorizations (e.g., Cholesky) that require $O(n^3)$ floating point operations (flops) to solve linear systems involving $\mathbf{C}$ and computing det($\mathbf{C}$). Finally, predicting the response at K new locations require an additional $O(Kn^2)$ flops. Alternative parametrizations such as integrating $\mathbf{w}$ out of (2) shrinks the size of the parameter space, but does not obviate these computational bottlenecks. Even for moderately large spatial datasets, say with with ~ 10^4–10^5 locations, these memory and storage demands become prohibitive. For the TIU dataset with 5 × 10^6 locations, implementing (2) is practically impossible.

As mentioned in the Introduction, we pursue massive scalability for full Bayesian inference exploiting the NNGP. The underlying idea is familiar in graphical models (see, e.g., Lauritzen, 1996; Murphy, 2012). The joint distribution for a random vector $\mathbf{w}$ can be looked upon as a directed acyclic graph (DAG). We write $p(\mathbf{w}) = p\left(w_1\right)\prod_{i=2}^{n} p\left(w_i|\text{Pa}[i]\right)$, where $w_i \equiv w(\mathbf{s}_i)$ and Pa[i] = $\{w_1, w_2,\ldots, w_{i-1}\}$ is the set of parents of w_i. We can construct sparse models for $\mathbf{w}$ by shrinking the size of Pa[i]. In spatial contexts, this can be done by defining Pa[i] to be the set of $w(\mathbf{s}_j)$'s corresponding to a small number m of nearest neighboring locations of $\mathbf{s}_i$. Approximations resulting from such shrinkage were originally proposed by Vecchia (1988) and studied and exploited by Stein et al. (2004); Stroud et al. (2017); Datta et al. (2016a,c); Huang and Sun (2018). The NNGP builds upon previous ideas and extends finite-dimensional likelihood approximations to well-defined sparsity-inducing Gaussian processes for estimating (2).

Working with multivariate Gaussian densities makes the connection between conditional independence in DAGs and sparsity abundantly clear. We can write the multivariate Gaussian density $N(\mathbf{w} \mid \mathbf{0}, \mathbf{C})$ as a linear model,

$$w_1 = 0 + \eta_1 \quad \text{and} \quad w_i = a_{i1}w_1 + a_{i2}w_2 + \cdots + a_{i,i-1}w_{i-1} + \eta_i \text{ for } i = 2, \ldots, n,$$

$w_1 = 0 + \eta_1$ and $w_i = a_{i1} w_1 + a_{i2} w_2 + \cdots + a_{i,i-1} w_{i-1} + \eta_i$ for $i = 2,\ldots, n$, or, more compactly, simply as $\mathbf{w} = \mathbf{A}\mathbf{w} + \boldsymbol{\eta}$, where $\mathbf{A}$ is $n \times n$ strictly lower-triangular with elements $a_{ij} = 0$ whenever $j \geq i$ and $\boldsymbol{\eta} \sim N(\mathbf{0}, \mathbf{D})$ and $\mathbf{D}$ is diagonal with entries $d_{11} = \text{var}(w_1)$ and $d_{ii} = \text{Var}(w_i | \{w_j : j < i\})$ for $i = 2,\ldots, n$.

From the structure of $\mathbf{A}$ it is evident that $\mathbf{I} - \mathbf{A}$ is nonsingular and $\mathbf{C} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{D}(\mathbf{I} - \mathbf{A})^{-\top}$, where for any matrix $\mathbf{M}$, $\mathbf{M}^{-\top}$ refers to the inverse of its transpose. For any matrix $\mathbf{M}$ and set of indices $I_1, I_2 \subseteq \{1, 2,\ldots, n\}$, let $\text{M}[I_1, I_2]$ denote the submatrix of $\mathbf{M}$ formed by the rows indexed by I_1 and columns indexed by I_2. With the addition of D[1,1] = C[1,1] and the first row of $\mathbf{A} = \mathbf{0}$, the calculation of $\mathbf{A}$ and $\mathbf{D}$ is given in Pseudocode 1, where 1:i denotes the set $\{1, 2,\ldots, i\}$, solve(B,b) computes the solution $\mathbf{x}$ for the linear system $\mathbf{Bx} = \mathbf{b}$, and dot(u,v) denotes the inner-product between two vectors $\mathbf{u}$ and $\mathbf{v}$.

Pseudocode 1: Computation of **A** and **D**.

```
for(i in 1:(n-1)) {
  A[i+1,1:i] = solve(C[1:i,1:i], C[1:i,i+1])
  D[i+1,i+1] = C[i+1,i+1] - dot(C[i+1,1:i],A[i+1,1:i])
}
```

While Pseudocode 1 computes the Cholesky decomposition of **C**, there is no apparent gain to be had from the preceding computations since, as the loop runs into higher values of i closer to n, the dimension of C[1:i,1:i] increases. Consequently, one will need to solve larger and larger linear systems and the computational complexity remains $O(n^3)$. Nevertheless, it immediately shows how to exploit sparsity if we set some elements in the lower triangular part of **A** to be zero. For example, suppose we permit no more than m elements in each row of **A** to be nonzero. Let N[i] be the set of indices j $<$ i such that A[i,j] $\neq$0. One can then compute the elements of **A** and **D** following Pseudocode 2.

Pseudocode 2: Sparsity inducing computation of **A** and **D**.

```
for(i in 1:(n-1)) {
  A[i+1,N[i+1]] = solve(C[N[i+1],N[i+1]], C[N[i+1],i+1])
  D[i+1,i+1] = C[i+1,i+1] - dot(C[i+1, N[i+1]], A[i+1,N[i+1]])
}
```

In Pseudocode 2 we solve n-1 linear systems of size at most m × m where $m = \max_i |\text{N(i)}|$.

This can be performed in $O(nm^3)$ flops. Furthermore, these computations can be performed in parallel as each iteration of the loop is independent of the others. The above discussion provides a very useful strategy for constructing a sparse precision matrix. Starting with a dense $n \times n$ matrix **C**, we construct a sparse strictly lower-triangular matrix **A** with no more than $m(\ll n)$ non-zero entries in each row, and the diagonal matrix **D** using Pseudocode 2 such that the matrix $\widetilde{\mathbf{C}} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{D}(\mathbf{I} - \mathbf{A})^{-\top}$ is a covariance matrix whose inverse $\widetilde{\mathbf{C}}^{-1} = (\mathbf{I} - \mathbf{A})^{\top}\mathbf{D}^{-1}(\mathbf{I} - \mathbf{A})$ is sparse. Figure 1 presents a visual representation of the sparsity.

The factorization of $\widetilde{\mathbf{C}}^{-1}$ facilitates cheap computation of quadratic forms $\mathbf{u}^{\top}\widetilde{\mathbf{C}}^{-1}\mathbf{v}$ in terms **A** and **D**. The algorithm to evaluate such quadratic forms qf(u,v,A,D) is provided in Pseudocode 3, where $*$ and / denote multiplication and division by scalars, respectively.

Observe the algorithm in Pseudocode 3 only involves inner products of $m \times 1$ vectors. So, the entire for loop can be computed using $O(nm)$ flops as compared to $O(n^2)$ flops typically required to evaluate quadratic forms involving an $n \times n$ dense matrix. Also, importantly, the determinant of $\widetilde{\mathbf{C}}$ is obtained with almost no additional cost—it is simply $\prod_{i=1}^{n} \mathrm{D}[i, i]$.

Pseudocode 3: Computation of quadratic form.

```
qf(u,v,A,D) = u[1] * v[1] / D[1,1]
for(i in 2:n) {
  qf(u,v,A,D) = qf(u,v,A,D) + (u[i] - dot(A[i,N(i)], u[N(i)]))
    ∗(v[i] - dot(A[i,N(i)], v[N(i)]))/D[i,i]
  }
```

Hence, while $\widetilde{\mathbf{C}}$ need not be sparse, the density $N(\mathbf{w} \mid \mathbf{0}, \widetilde{\mathbf{C}})$ is cheap to compute requiring only $O(n)$ flops. This was exploited by Datta et al. (2016a) where the neighbor sets were constructed based on m nearest neighbors and the traditional GP prior for **w** in (2) was replaced with an NNGP prior $N(\mathbf{w} \mid \mathbf{0}, \widetilde{\mathbf{C}})$. The Markov chain Monte Carlo (MCMC) implementation of the NNGP model in Datta et al. (2016a) requires updating the n latent spatial effects **w** sequentially, in addition to the regression and covariance parameters. While this ensures substantial computational scalability in terms of evaluating the likelihood, the behavior of MCMC convergence for such a high-dimensional model is difficult to study and may well prove unreliable.

We observed that, for very large spatial datasets, sequential updating of the random effects often leads to very poor mixing in the MCMC (see Figures S2 and S3). The computational gains per MCMC iteration is thus offset by a slow converging MCMC. Liu et al. (1994) showed that MCMC algorithms where one or more variables are marginalized out tend to have lower autocorrelation and improved convergence behavior. Here we explore NNGP models that drastically reduce the parameter dimensionality of the NNGP models by marginalizing over the entire vector of spatial random effects. Three different variants are developed, including an MCMC free conjugate model, and their relative merits and demerits are assessed both in terms of computational burden as well as model prediction and inference. Simulation experiments using spatial datasets of up to 10 million locations are conducted to assess the models' performance. Finally, we use the NNGP models to analyze the TIU dataset comprising over 5 million locations. To our knowledge, fully Bayesian analysis of spatial data at such scales is unprecedented.

2.1 Collapsed NNGP

The hierarchical model (2) or its NNGP analogue impart a nice interpretation to the spatial random effects. The latent surface $w(\mathbf{s})$ can provide a lot of information about the effect of missing covariates or unobserved physical processes. Hence, inference about $\mathbf{w}$ is often critical for the researchers in order to improve the understanding of the underlying scientific phenomenon. Here, we provide a collapsed NNGP model that enjoys the frugality of a low-dimensional MCMC chain but allows for full recovery of the latent random effects. We begin with the two-stage hierarchical specification $N(\mathbf{y} \mid \mathbf{X}\boldsymbol{\beta} + \mathbf{w}, \tau^2\mathbf{I}) \times N(\mathbf{w} \mid \mathbf{0}, \widetilde{\mathbf{C}})$ and avoid sampling $\mathbf{w}$ in the Gibbs' sampler by integrating out $\mathbf{w}$ to obtain the collapsed NNGP model

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta},\ \boldsymbol{\Lambda}) \text{ where } \boldsymbol{\Lambda} = \widetilde{\mathbf{C}} + \tau^2\mathbf{I} \quad (3)$$

This model has only $p + 4$ parameters compared to $n + p + 4$ parameters in the hierarchical model. We use a conjugate prior $N(\boldsymbol{\mu}_\beta, \mathbf{V}_\beta)$ for $\boldsymbol{\beta}$, Inverse Gamma priors for the spatial and noise variances, and uniform priors for the range and smoothness parameters. We use the $u \mid \cdot$ notation to denote the full conditional distribution of any random variable u in the Gibbs' sampler. Let $N(i)$ denote the set of indices corresponding to neighbor set of $\mathbf{s}_i$. Observe that, although from Section 2 we know $\widetilde{\mathbf{C}} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{D}(\mathbf{I} - \mathbf{A})^{-\top}$, $\boldsymbol{\Lambda}$ does not enjoy any such convenient factorization. In fact, $\boldsymbol{\Lambda}^{-1}$ is also not guaranteed to be sparse, but exploiting the Sherman Woodbury Morrison (SWM) identity, we can write $\boldsymbol{\Lambda}^{-1} = \boldsymbol{\tau}^{-2}\mathbf{I} - \boldsymbol{\tau}^{-4}\boldsymbol{\Omega}^{-1}$ where $\Omega = \left(\widetilde{\mathbf{C}}^{-1} + \tau^{-2}\mathbf{I}\right)$ enjoys the same sparsity as $\widetilde{\mathbf{C}}^{-1}$. Also, using a familiar determinant identity, we have $\det(\boldsymbol{\Lambda}) = \tau^{2n}\det(\widetilde{\mathbf{C}})\det(\boldsymbol{\Omega})$.

We exploit these matrix identities in conjunction with sparse matrix algorithms to obtain posterior distributions of the parameters $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\tau}^2\}$. In fact, the necessary computations can be done by entirely avoiding expensive matrix computations and is described in detail in Algorithm 1. In addition to the inner product function dot(·, ·) introduced earlier, we require a fill-reducing permutation matrix and a sparse Cholesky factorization (sparsechol(·)) for a sparse positive-definite matrix (note, dot(·, ·), sparsechol(·), and subsequent functions that share this font are pseudocode). Large matrix-matrix and matrix-vector multiplications either involve at least one triangular matrix (trmm(·, ·) or trmv(·, ·), where mm and mv denote matrix-matrix and matrix-vector operations) or at least one sparse matrix (sparsemm(·, ·) or sparsemv(·, ·)). We also use diagsolve(·, ·) and trsolve(·, ·) to solve linear systems with a diagonal or triangular coefficient matrix, respectively. We perform Cholesky decompositions, matrix-vector multiplications and solve linear equations involving general unstructured matrices using chol(·), gemv(·, ·) and solve(·, ·), respectively, only for small $p \times p$ or $m \times m$ matrices where both p and m are much less than n. Other utilities used in Algorithm 1 are diag(·) to extract the diagonal elements of a matrix, prod(·) to compute the product of the elements in a vector and rnorm(·) to generate a specified number of random variables (as an integer argument) from a standard $N(0, 1)$ distribution. Algorithm 1 Collapsed NNGP: Sampling from the posterior

```
MCMC steps for updating $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\}$
1: Gibb's sampler update for $\boldsymbol{\beta}$:
   $\boldsymbol{\beta} \mid \cdot \sim N\left(\mathbf{B}^{-1}\mathbf{b}, \mathbf{B}^{-1}\right)$, where $\mathbf{B} = \mathbf{X}^\top \boldsymbol{\Lambda}^{-1}\mathbf{X} + \mathbf{V}_\beta^{-1}$ and $\mathbf{b} = \mathbf{X}^\top \boldsymbol{\Lambda}^{-1}\mathbf{y} + \mathbf{V}_\beta^{-1}\boldsymbol{\mu}_\beta$
     (a) Use Pseudocode 1 to obtain A and D using C and $\{N(i) \mid i = 1, 2, \ldots, n\}$        $O\left(nm^3\right)$ flops
     (b) $\Omega = \mathrm{trmm}\left((\mathrm{I} - \mathrm{A})^\top, \mathrm{diagsolve}(\mathrm{D}, \mathrm{I} - \mathrm{A})\right) + \tau^{-2} * \mathrm{I}$        $O\left(nm^2\right)$flops
     (c)Find a fill reducing permutation matrix P for $\Omega$
     (d) $\mathrm{L} = \mathrm{sparsechol}\ (\mathrm{sparsemm}\ (\mathrm{sparsemm}(\mathrm{P}, \Omega), \mathrm{P}^\top))$
     (e) for (j in 1:n) {
             $\mathrm{u}_j = \mathrm{trsolve}\ (\mathrm{L}, \mathrm{sparsemv}\ (\mathrm{P}, \mathrm{X}[, \mathrm{j}]))$;        $\mathrm{v}_j = \mathrm{trsolve}\left(\mathrm{L}^\top, \mathrm{u}_j\right)$
         }
     (f) $\mathrm{F} = \mathrm{solve}\left(\mathrm{V}_\beta, \mathrm{I}\right)$;   $\mathrm{f} = \mathrm{solve}\left(\mathrm{V}_\beta, \boldsymbol{\mu}_\beta\right)$        $O\left(p^3\right)$flops
     (g) Solve for $p \times p$ matrix $\mathbf{B}$ and $p \times 1$ vector $\mathbf{b}$:        $O\left(np^2\right)$flops
         for (j in 1:p) {
             $\mathrm{b}[\mathrm{j}] = \mathrm{dot}\ (\mathrm{y}, \mathrm{X}[, \mathrm{j}])/\tau^2 - \mathrm{dot}\left(\mathrm{y}, \mathrm{sparsemv}\left(\mathrm{P}, \mathrm{v}_j\right)\right)/\tau^4 + \mathrm{f}[\mathrm{j}]$
             for (i in 1:p) {
                 $\mathrm{B}[\mathrm{i}, \mathrm{j}] = \mathrm{dot}(\mathrm{X}[, \mathrm{i}], \mathrm{X}[, \mathrm{j}])/\tau^2 - \mathrm{dot}\left(\mathrm{X}[, \mathrm{i}], \mathrm{sparsemv}\left(\mathrm{P}, \mathrm{v}_j\right)\right)/\tau^4 + \mathrm{F}[\mathrm{i}, \mathrm{j}]$
             }
         }
     (h) $\boldsymbol{\beta} = \mathrm{solve}(\mathrm{B}, \mathrm{b}) + \mathrm{trsolve}(\mathrm{chol}\ (\mathrm{B}), \mathrm{rnorm}\ (\mathrm{p}))$        $O\left(p^3\right)$flops
2: Metropolis – Hastings (MH) update for $\left\{\boldsymbol{\theta}, \tau^2\right\}$
   $p\left(\boldsymbol{\theta}, \tau^2 \mid \cdot\right) \propto p\left(\boldsymbol{\theta}, \tau^2\right) \times \frac{1}{\sqrt{\det(\boldsymbol{\Lambda})}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top \boldsymbol{\Lambda}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right)$
     (a) $\mathrm{r} = \mathrm{y} - \mathrm{gemv}\ (\mathrm{X}, \boldsymbol{\beta})$; $\mathrm{u} = \mathrm{trsolve}(\mathrm{L}, \mathrm{sparsemv}(\mathrm{P}, \mathrm{r}))$; $\mathrm{v} = \mathrm{trsolve}\left(\mathrm{L}^\top, \mathrm{u}\right)$        $O(np)$flops
     (b) $\mathrm{q} = \mathrm{dot}\ (\mathrm{r}, \mathrm{r})/\tau^2 - \mathrm{dot}\ (\mathrm{r}, \mathrm{sparsem}(\mathrm{P}, \mathrm{v}))/\tau^4$
     (c) $\mathrm{d} = \tau^{2 * n} * \mathrm{prod}\ (\mathrm{diag}\ (\mathrm{D})) * \mathrm{prod}\ (\mathrm{diag}\ (\mathrm{L}))^2$
     (d) Generate $\mathrm{p}\left(\boldsymbol{\theta}, \tau^2 \mid \cdot\right) \propto \frac{\exp(-\mathrm{q}/2) * \mathrm{p}\left(\boldsymbol{\theta}, \tau^2\right)}{\mathrm{sqrt}(\mathrm{d})}$
3: Repeat Steps(1)and(2)(except Step1(c)) $N$ times to obtain $N$ MCMC samples for $\left\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\right\}$
```

Observe that the entire Algorithm 1 is devoid of any expensive operations like solve, chol or gemv on dense $n \times n$ matrices. All such operations are limited to $m \times m$ or $p \times p$ matrices, where both m and p are small. The computational costs in terms of flops of all such steps are listed in the algorithm and are linear in n. However, the exact cost of the steps involving L in Algorithm 1 (Steps 1(c)-(e)) depends on the data design. Although Ω is sparse $O(nm^2)$ non-zero entries, the sparsity of its Cholesky factor L actually depends on the location of the non-zero entries. Hence we used a fill reducing permutation P that increases the sparsity of the Cholesky factor. Although P needs to be evaluated only once before the MCMC, finding the optimal P yielding the least fill-in is an NP-complete problem. Hence algorithms have been proposed to improve sparsity patterns based on a variety of fill-in minimizing heuristics, see, e.g., Amestoy et al. (1996), Karypis and Kumar (1998), Hager (2002) (also see Section 3).

When flops per iteration of MCMC are considered, computational requirements for the collapsed NNGP model is data dependent and may exceed the exact linear flops usage for the hierarchical NNGP Algorithm. We also observed this in simulation experiments described in Section 3. However, the improved MCMC convergence for the collapsed NNGP, as observed in Figures S2 and S5, implies that substantial computational gains accrue by truncating the MCMC run. Furthermore, all the for loops in Algorithm 1 can be evaluated independent of each other using parallel computing resources.

The collapsed model nicely separates the MCMC sampler for parameter estimation from posterior estimation of spatial random effects and subsequent predictions. Computational benefits accrue from using the quantities **L** and **u** already computed in Steps 1(d) and 2(a) of Algorithm 1 corresponding to the post-convergence samples of $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\}$. This is presented in the algorithm below. Algorithm 2

Collapsed NNGP: Posterior predictive inference

```
Post − MCMC steps using L and u from Steps 1(d) and 2(a) of Algorithm 1 for post −
  convergence samples of $\left\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\right\}$
1: Sample from $p(\mathbf{w} \mid \cdot)$ one − for − one for each post − convergence sample of $\left\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\right\}$
   $\mathbf{w} \mid \cdot \sim N(\mathbf{B}^{-1}\mathbf{b}, \mathbf{B}^{-1})$, where $\mathbf{B} = \widetilde{\mathbf{C}}^{-1} + \tau^{-2}\mathbf{I}$ and $\mathbf{b} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})/\tau^2$
    (a) $\mathrm{z} = \mathrm{rnorm}(n)$                                          $O(n)$Aops
    (b) $\mathrm{w} = \mathrm{sparsemv}\left(\mathrm{P}^\top, \mathrm{trsolve}\left(\mathrm{L}^\top, \mathrm{u}/\tau^2 + \mathrm{z}\right)\right)$
2: Prediction at a new location $\mathrm{s}_0$:
   $y(\mathbf{s}_0) \mid \cdot \sim N(\mathbf{x}(\mathbf{s}_0)^\top \boldsymbol{\beta} + w(\mathbf{s}_0), \tau^2)$
    (a) Find $\mathrm{N}_0 -$ set of $m$ nearest neighbors of $\mathbf{s}_0$ among $\left\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\right\}$      $O(n)$ flops
    (b) $\mathrm{c} = \mathrm{C}\left(\mathrm{s}_0, \mathrm{N}_0; \boldsymbol{\theta}\right)$                              $O(m)$ flops
    (c) $\mathrm{m} = \mathrm{dot}\left(\mathrm{c}, \mathrm{solve}\left(\mathrm{C}\left(\mathrm{N}_0, \mathrm{N}_0\right), \mathrm{w}\left[\mathrm{N}_0\right]\right)\right)$              $O\left(m^3\right)$ flops
        $\mathrm{v} = \mathrm{C}\left(\mathrm{s}_0, \mathrm{s}_0; \boldsymbol{\theta}\right) - \mathrm{dot}\left(\mathrm{c}, \mathrm{solve}\left(\mathrm{C}\left(\mathrm{N}_0, \mathrm{N}_0\right), \mathrm{c}\right)\right)$
    (d) $\mathrm{w}\left(\mathrm{s}_0\right) = \mathrm{m} + \mathrm{sqrt}(\mathrm{v}) * \mathrm{rnorm}(1)$                 $O(p)$ flops
        $\mathrm{y}\left(\mathrm{s}_0\right) = \mathrm{dot}\left(\mathrm{x}\left(\mathrm{s}_0\right), \boldsymbol{\beta}\right) + \mathrm{w}\left(\mathrm{s}_0\right) + \tau * \mathrm{rnorm}(1)$      $O(p)$ flops
```

Algorithm 2 demonstrates how inference on $w(\mathbf{s})$ and $y(\mathbf{s})$ can be easily achieved for any spatial location using the post burn-in samples of $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\}$. We first sample the spatial random effects $p(\mathbf{w} \mid \mathbf{y})$ for the observed locations, use them to sample from $p(w(\mathbf{s}_0) \mid \mathbf{y})$ and then from $p(y(\mathbf{s}_0) \mid \mathbf{y})$.

2.2 NNGP for the response

Both the sequential NNGP Algorithm in Datta et al. (2016a) or the collapsed version in Section 2.1 accomplishes prediction at a new location via recovering the spatial random effects first, proceeded by kriging at the new location. This differed from Vecchia (1988)'s original approach which applied nearest neighbor approximation directly to the marginal likelihood of y. The recovery of the spatial random effects becomes necessary if inference on the latent process is of interest. Although recovering w, as discussed earlier, has its own importance, if spatial interpolation of the response is the primary objective, this intermediate step is often a computational burden. In this Section, we propose a NNGP model for the response y that sacrifices the ability to recover w and directly predicts the response at new locations.

Datta et al. (2016a) demonstrated that a NNGP model can be derived from any Gaussian Process. If $w(\mathbf{s}) \sim GP(0, C(\cdot, \cdot))$ then the response $y(\mathbf{s}) \sim GP(\mathbf{x}(\mathbf{s})^\top \boldsymbol{\beta}, \Sigma(\cdot, \cdot))$ is also a Gaussian Process where $\Sigma(\mathbf{s}_i, \mathbf{s}_j) = C(\mathbf{s}_i, \mathbf{s}_j) + \tau^2 I(\mathbf{s}_i = \mathbf{s}_j)$. Hence, we can directly derive an NNGP for the response process $y(\mathbf{s})$. For finite dimensional realizations $\mathbf{y}$, likelihood under the response NNGP model is identical to Vecchia's composite likelihood. Datta et al. (2016a) extend this notion to a fully Bayesian setup. The key observation is that Vecchia's approximation corresponds to a proper multivariate Gaussian distribution obtained by simply replacing the covariance matrix $\boldsymbol{\Sigma} = \mathbf{C} + \boldsymbol{\tau}^2\mathrm{I}$ with its nearest-neighbor approximation $\widetilde{\Sigma}$ as described in Section 2. The sparsity properties documented in Section 2 apply to $\widetilde{\Sigma}$ as well. MCMC steps for parameter estimation and prediction using this response NNGP model are provided in Algorithm 3. Algorithm 3

Response NNGP model: Sampling from the posterior

```
MCMC steps for updating $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\}$
1:  Gibb's sampler update for $\boldsymbol{\beta}$:
    $\boldsymbol{\beta} \mid \cdot \sim N(\mathbf{B}^{-1}\mathbf{b}, \mathbf{B}^{-1})$, where  $\mathbf{B} = \mathbf{X}^\top\widetilde{\boldsymbol{\Sigma}}^{-1}\mathbf{X} + \mathbf{V}_\beta^{-1}$ and $\mathbf{b} = \mathbf{X}^\top\widetilde{\boldsymbol{\Sigma}}^{-1}\mathbf{y} + \mathbf{V}_\beta^{-1}\boldsymbol{\mu}_\beta$
      (a) Use Pseudocode 1 to obtain A and D using $\sum$ and $\{N(i) \mid i = 1, 2, \ldots, n\}$        $O(nm^3)$ flops
      (b) F = solve $(\mathrm{V}_\beta, \mathrm{I})$;  f = solve $(\mathrm{V}_\beta, \boldsymbol{\mu}_\beta)$        $O(p^3)$flops
      (c) Solve for $p \times p$ matrix $\mathbf{B}$ and $p \times 1$ vector $\mathbf{b}$ using Pseudocode 3        $O(nmp^2)$flops
          for  (i in 1:p)  {
              b[i] = qf(X[, i], y, A, D) + f[$i$]
              for  (j in 1:p)  {
                  B [1 , j] = qf (X [, i], X [, j], A , D) + F [1, $j$]
              }
          }
      (d) $\boldsymbol{\beta}$ = solve (B, b) + trsolve ( chol (B), rnorm (p))        $O(p^3)$ flops
2:  Metropolis – Hastings (MH) update for $\{\boldsymbol{\theta}, \tau^2\}$:
    $p(\boldsymbol{\theta}, \tau^2 \mid \cdot) \propto p(\boldsymbol{\theta}, \tau^2) \times \frac{1}{\sqrt{\det(\widetilde{\boldsymbol{\Sigma}})}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top \widetilde{\boldsymbol{\Sigma}}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right)$
      (a) e = y − gemv (X, $\boldsymbol{\beta}$);  Using Pseudocode 3,  q = qf(e, e, A, D)        $O(n(p + m))$ flop
      (b) d = prod ( diag (D))        $O(n)$ flops
      (c) Generate p$(\boldsymbol{\theta}, \tau^2 \mid \cdot) \propto \frac{\exp(-\mathrm{q}/2) * \mathrm{p}(\boldsymbol{\theta}, \tau^2)}{\mathrm{sqrt}\ (\mathrm{d})}$        $O(1)$flops
3:  Repeat Steps (1) and (2) $N$ times to obtain $N$ MCMC samples for $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\}$
```

Unlike the collapsed NNGP model, the computational cost for each step of Algorithm 3 does not depend on the spatial design of the data and is exactly linear in n. This is a result of the complete absence of the latent spatial effects w in the model. Once again, parallel computing can be leveraged to evaluate all the for loops. A caveat with the response model is that recovery of w is not possible as highlighted in Datta et al. (2016a). However, if that is of peripheral concern, the response model offers a computationally parsimonious solution for fully Bayesian analysis of massive spatial datasets. Posterior predictive inference, therefore, consists only of predicting the outcome y(s) at any arbitrary location s. This is achieved easily through Algorithm 4 given below, where $y_{N(s0)}$ represents the subvector of y corresponding to the points in N(s$_0$), $\mathbf{X}_{N(\mathbf{s}_0)}$ is the corresponding design matrix, and Σ_0 is the $m \times m$ covariance matrix for $\mathbf{y}_{N(\mathbf{s}_0)}$. Algorithm 4

Response NNGP model: Posterior predictive inference

```
**Post − MCMC steps using post − convergence samples of** $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\}$
1: **Sample from** $p(y(\mathbf{s}_0) \mid \cdot)$ **one − for − one for each post − convergence sample of** $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\}$
    $y(\mathbf{s}_0) \mid \cdot \sim N\ (\mathbf{x}(\mathbf{s}_0)^\top \boldsymbol{\beta} + \mathbf{c}_0^\top \Sigma_0^{-1} (\mathbf{y}_{N(\mathbf{s}_0)} - \mathbf{X}_{N(\mathbf{s}_0)} \boldsymbol{\beta}),\ \sum(\mathbf{s}_0, \mathbf{s}_0) - \mathbf{c}_0^\top \Sigma_0^{-1} \mathbf{c}_0)$
      (a) Find $\mathrm{N}_0 -$ set of $m$ nearest neighbors of $\mathrm{s}_0$ among $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$        $O(n)$ flops
      (b) $\mathrm{c} = \Sigma(\mathrm{s}_0, \mathrm{N}_0; \boldsymbol{\theta})$        $O(m)$ flops
      (c) $\mathrm{m} = \mathrm{dot(c, solve}(\sum[\mathrm{N}_0, \mathrm{N}_0], \mathrm{y}[\mathrm{N}_0] - \mathrm{dot}(\mathrm{X}[\mathrm{N}_0,], \boldsymbol{\beta}))$        $O(m^3)$ flops
          $\mathrm{v} = \Sigma(\mathrm{s}_0, \mathrm{s}_0) - \mathrm{dot}(\mathrm{c}, \mathrm{solve}(\Sigma[\mathrm{N}_0, \mathrm{N}_0], \mathrm{c}))$
      (d) $\mathrm{y}(\mathrm{s}_0) = \mathrm{dot}(\mathrm{x}(\mathrm{s}_0), \boldsymbol{\beta}) + \mathrm{m} + \mathrm{sqrt(v)} * \mathrm{rnorm}(1)$        $O(n)$ flops
```

2.3 MCMC-free exact Bayesian inference using conjugate NNGP

The fully Bayesian approaches developed in Datta et al. (2016a) and in Sections 2.1 and 2.2 provide complete posterior distributions for all parameters. However, for massive spatial datasets containing millions of observations, running the Gibbs' samplers for several thousand iterations may still be prohibitively slow. One advantage of NNGP over similar scalable statistical approaches for large spatial data is that it offers a probability model. Here, we exploit this fact to achieve exact Bayesian inference.

We define $\alpha = \tau^2/\sigma^2$ and rewrite the marginal model from Section 2.2 as $N(\mathbf{y} \mid \mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{M})$, where $\mathbf{M} = \mathbf{G} + \alpha\mathbf{I}$ and $\mathbf{G}$ denotes the Matern correlation matrix corresponding to the covariance matrix $\mathbf{C}$ i.e. $\mathbf{G}[i, j] = C(\mathbf{s}_i, \mathbf{s}_j, (1, \nu, \phi)^\top)$. Once again, the analogous NNGP model can be obtained by replacing the dense matrix $\mathbf{M}$ with its nearest-neighbor approximation $\widetilde{\mathbf{M}}$. Note that $\widetilde{\mathbf{M}}$ depends on α, the spatial range ϕ and smoothness ν. Empirically, in spatial regression models, the spatial process parameters ϕ and ν are often not well estimated due to multimodality issues. In fixed domain asymptotic settings (see, e.g., Zhang, 2004) it is impossible to jointly identify the spatial covariance parameters. Consequently, if inference for the covariance parameters is not of interest, it might be possible to fix them at reasonable values with minimal effect on prediction or point estimates of other model parameters. For example, the smoothness parameter ν could be fixed at 0.5, which reduces(1) to the exponential covariance function, and ϕ and α could be estimated using K-fold cross-validation.

For fixed α and ϕ, we obtain the familiar conjugate Bayesian linear regression model $IG\left(\sigma^2 \mid a_\sigma, b_\sigma\right) \times N(\boldsymbol{\beta} \mid \boldsymbol{\mu}_\beta, \sigma^2 \mathbf{V}_\beta) \times N(\mathbf{y} \mid \mathbf{X}\boldsymbol{\beta}, \sigma^2 \widetilde{\mathbf{M}})$ with joint posterior distribution

$$p\left(\boldsymbol{\beta}, \sigma^2 \mid \mathbf{y}\right) \propto \underbrace{IG\left(\sigma^2 \mid a_\sigma^*, b_\sigma^*\right)}_{p\left(\sigma^2 \mid \mathbf{y}\right)} \times \underbrace{N(\boldsymbol{\beta} \mid \mathbf{B}^{-1}\mathbf{b}, \sigma^2 \mathbf{B}^{-1})}_{p(\boldsymbol{\beta} \mid \sigma^2, \mathbf{y})},$$

where $a_\sigma^* = a_\sigma + n/2, b_\sigma^* = b_\sigma + \frac{1}{2}\left(\boldsymbol{\mu}_\beta^\top \mathbf{V}_\beta^{-1} \boldsymbol{\mu}_\beta + \mathbf{y}^\top \widetilde{\mathbf{M}}^{-1} \mathbf{y} - \mathbf{b}^\top \mathbf{B}^{-1} \mathbf{b}\right)$, $\mathbf{B} = \mathbf{V}_\beta^{-1} + \mathbf{X}^\top \widetilde{\mathbf{M}}^{-1} \mathbf{X}$ and $\mathbf{b} = \mathbf{V}_\beta^{-1} \boldsymbol{\mu}_\beta + \mathbf{X}^\top \widetilde{\mathbf{M}}^{-1} \mathbf{y}$ It is easy to directly sample $\sigma^2 \sim IG\left(a_\sigma^*, b_\sigma^*\right)$ and then sample $\boldsymbol{\beta} \sim N(\mathbf{B}^{-1}\mathbf{b}, \sigma^2 \mathbf{B}^{-1})$ one-for-one for each drawn σ^2. This produces samples from the marginal posterior distributions $\boldsymbol{\beta} \mid \mathbf{y} \sim \text{MVS-}t_{2a_\sigma^*}\left(\mathbf{B}^{-1}\mathbf{b}, \frac{b_\sigma^*}{a_\sigma^*}\mathbf{B}^{-1}\right)$ and $\sigma^2 \mid \mathbf{y} \sim IG\left(a_\sigma^*, b_\sigma^*\right)$ where $\text{MVS-}t_\kappa(\mathbf{B}^{-1}\mathbf{b}, (b/a)\mathbf{B}^{-1})$ denotes the multivariate non-central Student's t distribution with degrees of freedom κ, mean $\mathbf{B}^{-1}\mathbf{b}$ and variance $b\mathbf{B}^{-1}/(a - 1)$. The marginal posterior mean and variance for σ^2 are $b_\sigma^*/\left(a_\sigma^* - 1\right)$ and $b_\sigma^{*2}/\left(a_\sigma^* - 1\right)^2\left(a_\sigma^* - 2\right)$, respectively.

Instead of sampling from the posterior directly, we prefer a fast evaluation of the marginal posterior distributions to effectively implement the aforementioned cross-validatory approach. Steps for efficiently evaluating the above is provided in Algorithm 5. The marginal posterior predictive distribution at a new location $\mathbf{s}_0$ is given by $y\left(\mathbf{s}_0\right) \mid \mathbf{y} \sim t_{2a_\sigma^*}\left(m_0, b_\sigma^* v_0 / a_\sigma^*\right)$, where expressions for m_0 and v_0 are provided in Step 3 of Algorithm 5. We deploy hyper-parameter tuning based on K-fold cross-validation to choose the optimal α and ϕ from a grid of possible values. In our data analysis, we have chosen broad endpoints of the grid using exploratory variograms. However, as suggested by one reviewer, reparametrizing $\alpha^* = \alpha/(1 + \alpha)$ and $\phi^* = \phi/(1 + \phi)$ would ensure that the new hyper-parameters are within [0, 1] and can facilitate a more automated grid-search. In applications, where the exploratory variograms are inaccurate, the latter parametrization will possibly be more useful.

We denote the indices and locations corresponding to the k-th fold of the data by $I(k)$ and $S(k)$ respectively whereas $I(-k)$ and $S(-k)$ respectively denote the analogous quantities when the k^{th} fold is excluded from the data. Also, let $N(i, k)$ denote the neighbor set for a location $\mathbf{s}_i$ constructed from the locations in $S(-k)$. Details of the cross-validation procedure are also provided in Algorithm 5. Algorithm 5

MCMC free posterior sampling for conjugate NNGP model

Hyper parameter tuning

```
1: Fix $\alpha$ and $\phi$, split the data into $K$ folds.
    (a) Find the collection of neighbor sets $\mathcal{N} = \{N(i,k): i = 1,2,\ldots,n; k = 1,2,\ldots,K\}$
2: Obtain posterior means for $\boldsymbol{\beta}$ and $\sigma^2$ after removing the $k^{th}$ fold of the data:
    (a) Use Pseudocode 1 to obtain A(k) and D(k) from $\mathbf{M}[S(-k), S(-k)]$ and $\mathcal{N}$    $O\left(nm^3\right)$ flops
    (b) $\mathrm{F} = \mathrm{solve}\left(\mathrm{V}_\beta, \mathrm{I}\right)$;  $\mathrm{f} = \mathrm{solve}\left(\mathrm{V}_\beta, \boldsymbol{\mu}_\beta\right)$    $O\left(p^3\right)$ flops
    (c) Solve for $p \times p$ matrix $\mathbf{B}(k)$ and $p \times 1$ vector $\mathbf{b}(k)$ using Pseudocode 3:    $O\left(nmp^2\right)$ flops
        for (i in 1:p) {
            b(k) [i] = qf(X[S(−k), i], y[S(−k)], A(k), D(k)) + f[i]
            for (j in 1:p) {
                B(k)[i, j] = qf (X[S(−k), i], X[S(−k), j], A (k), D(k)) + F[i, j]
            }
        }
    (d) $\mathrm{V(k)} = \mathrm{solve}\ (\mathrm{B}(k), \mathrm{I})$; $\mathrm{g}(k) = \mathrm{gemv}\ (\mathrm{V(k)}, \mathrm{b(k)})$    $O\left(p^3\right)$ flops
        $\mathrm{a}_\sigma^*(\mathrm{k}) = \mathrm{a}_\sigma + (\mathrm{n} - \mathrm{n/K})/2$
        $\mathrm{b}_\sigma^*(\mathrm{k}) = \mathrm{b}_\sigma + \left(\mathrm{dot}\left(\boldsymbol{\mu}_\beta, \mathrm{f}\right) + \mathrm{qf(y[S(-k)], y[S(-k)], A(k), D(k))} - \mathrm{dot\ (g(k), b(k))}\right)/2$
    (e) $\widehat{\boldsymbol{\beta}} = \mathrm{g}(k)$; $\widehat{\sigma^2} = \mathrm{b}_\sigma^*(\mathrm{k})/\left(\mathrm{a}_\sigma^*(\mathrm{k}) - 1\right)$
3: Predicting posterior means of $\mathbf{y}[S(k)]$:    $O\left(nm^3/K\right)$ flops
        for (s in S(k)) {
            N(s, k) = m-nearest neighbors of $\mathbf{s}$ from $\mathbf{S}(-k)$
            $\mathrm{z} = \mathrm{M(s, N(s,k))}$
            $\mathrm{w} = \mathrm{solve\ (M[N(s,k), N(s,k)], z)}$
            $\mathrm{m}_0 = \widehat{\mathrm{y(s)}} = \mathrm{dot\ (x(s),\ \ g(k))} + \mathrm{dot\ (w, (y[N(s,k)]} - \mathrm{dot\ (X[N(s,k), ], g(k))))}$
            $\mathrm{u} = \mathrm{x(s)} - \mathrm{dot\ (X[N(s,k), ], w)}$
            $\mathrm{v}_0 = \mathrm{dot\ (u,\ gem\ v(V(k), u))} + 1 + \alpha - \mathrm{dot\ (w, z)}$
            $\mathrm{Var}\ \widehat{\mathrm{(y(s))}} = \mathrm{b}_\sigma^*(\mathrm{k})\mathrm{v}_0/\left(\mathrm{a}_\sigma^*(\mathrm{k}) - 1\right)$
        }
4: Root Mean Square Predictive Error (RMSPE) over $K$ folds:    $O(n)$flops
    (a) Initialize $e = 0$
        for (k in 1:K) for $\left(\mathrm{s}_i \text{ in } \mathrm{S[k]}\right)$ {
            $\mathrm{e} = \mathrm{e} + \left(\mathrm{y}\left(\mathrm{s}_i\right) - \widehat{\mathrm{y}\left(\mathrm{s}_i\right)}\right)^2$
        }
5: Cross validation for choosing $\alpha$ and $\phi$
    (a) Repeat steps (2) and (3) for $G$ values of $\alpha$ and $\phi$    $O\left(GKnm\left(p^2 + m^2\right)\right)$ flops
    (b) Choose $\alpha_0$ and $\phi_0$ as the value that minimizes the average RMSPE    $O(G)$ flops
```

Parameter estimation and prediction

```
6: Repeat step (2) with $\left(\alpha_0, \phi_0\right)^\top$ and the full data to get $\left(\boldsymbol{\beta}, \sigma^2\right) \mid \mathbf{y}$    $O\left(nmp^2 + nm^3\right)$flops
7: Repeat step (3) with $\left(\alpha_0, \phi_0\right)^\top$ and the full data to predict at a new location $\mathbf{s}_0$ to obtain the
    mean and variance of $y\left(\mathbf{s}_0\right) \mid \mathbf{y}$    $O\left(m^3\right)$ flops
```

Algorithm 5 completely circumvents MCMC based iterative sampling and only requires at most $O(n)$ flops per step. Although the calculations need to be replicated for every (ϕ, α) combination, unlike the MCMC based algorithms that run serially, this step can be run in parallel. Moreover, kriging is often less sensitive to the choice of the covariance parameters so cross-validation can be done at a moderately crude resolution on the (ϕ, α) domain. Hence, the Algorithm remains extremely fast. This incredible scalability makes the conjugate NNGP model an attractive choice for ultra high-dimensional spatial data. Although this approach philosophically departs from the true Bayesian paradigm, often inference about covariance parameters is of little interest and this hybrid cross-validation approach offers a pragmatic compromise.

3 Illustrations

3.1 Implementation

This section details two simulation experiments and the analysis of a large remotely sensed dataset. In the analyses, we consider the candidate models labeled: *Sequential* defined in Datta et al. (2016a); *Collapsed* defined in Section 2.1; *Response* defined in Section 2.2, and; *Conjugate* defined in Section 2.3.

Two additional analyses are provided in the web supplement. The first, Section S3, compares full GP and NNGP model parameter estimates and predictive performance. The second, Section S4, moves beyond the typical geostatistical setting where $\mathbf{s}$ indexes data in two-dimensions, e.g., latitude and longitude, to a more general settings where data are indexed in N-dimensions. Such data are common in computer experiments, where $\mathbf{s}$ indexes outcomes associated with a set of values on N computer model inputs. Here too, we apply a Matérn covariance function. Response and Conjugate model out-of-sample predictive performance is shown to be comparable with that achieved using a local approximate Gaussian processes as implemented in the laGP R package (Gramacy and Sun, 2017; Gramacy, 2016). Samplers were programmed in C++ and used openBLAS (Zhang, 2016) and Linear Algebra Package (LAPACK; www.netlib.org/lapack) for efficient matrix computations. openBLAS is an implementation of Basic Linear Algebra Subprograms (BLAS; www.netlib.org/blas) capable of exploiting multiple processors. Additional multiprocessor parallelization used openMP (Dagum and Menon, 1998) to improve performance of key steps within the samplers. In particular, substantial gains were realized by distributing the calculation of NNGP precision matrix components using the openMP omp for directive. Updating these matrices is necessary for each MCMC iteration in the Sequential, Response, and Collapsed models, and for each Conjugate model cross-validation iteration. An omp for directive with reduction clause was also effectively used to evaluate the Pseudocode 3 found in all models.

For the Collapsed model, SuiteSparse version 4.4.5 (Davis, 2016a) provided an interface to: fill-in minimizing algorithms, e.g., AMD (Amestoy et al., 2004) and METIS (Karypis and Kumar, 1998); CHOLMOD (Chen et al., 2008) version 3.0.6 used for supernodal openBLAS-based Cholesky factorization to obtain $\mathbf{L}$ of $\mathbf{P}\left(\widetilde{\mathbf{C}}^{-1} + \tau^{-2}\mathbf{I}\right)\mathbf{P}^{\top}$, and solvers for sparse triangular systems. Also see the text by Davis (2006).

For each analysis using the Collapsed model, nine fill-in algorithms were considered (for details see Chen et al., 2008; Davis, 2016b, pages 4 and 16, respectively) for formation of the permutation matrix **P**. Assessment of the various fill-in algorithms is based on the resulting pattern of non-zero matrix elements. This is important for our setting because the initial pattern of the NNGP precision matrix is determined by the neighbor set and, hence, discovery of an *optimal* permutation matrix need only be done once prior to sampling.

Implementing NNGP models requires a neighbor set for each observed location. For a given location $\mathbf{s}_i$, a brute force approach to finding the neighbor set calculates Euclidean distances to $\mathbf{s}_1$, $\mathbf{s}_2$ and $\mathbf{s}_{i-1}$, sorts these distances while keeping track of locations' indexes, then selects the m minimum distance neighbors. This brute force approach is computationally demanding. Subsequent analyses use a relatively simple to implement fast nearest neighbor search algorithm proposed by Ra and Kim (1993) that provides substantial efficiency gains over the brute force search (see supplemental material for details).

All subsequent analyses were conducted on a Linux workstation with two 18-core Intel processors and 512 GB of memory. Unless otherwise noted, posterior inference used the last 1×10^4 iterations from each of three chains of 2.5×10^4 iterations. Chains run for a given model were initiated at different values and each chain was given a unique random number generator seed. Following Datta et al. (2016a), all models were fit using m=15 neighbors unless noted otherwise.

Code and data needed to reproduce the analyses are provided in the web supplement.

3.2 Experiment #1

The aim of this experiment was to assess NNGP model run time. To achieve this, we selected data subsets for a range of n from the TIU dataset described in Sections 1 and 3.4. The posited model follows (2) and includes an intercept and slope regression coefficients, and an exponential covariance function with parameters σ^2, ϕ, and residual variance τ^2. A "flat" improper prior distribution was assigned to each regression coefficient, β's, which places equal weight on all possible values of the parameter. The variance components τ^2 and σ^2 were assigned inverse-Gamma $IG(2, 10)$ priors, and a uniform $U(0.1, 10)$ prior for the decay parameter ϕ. The support on the decay corresponds to an effective spatial range (i.e., the distance where the spatial correlation is 0.05) between 0.3 to 30 km (see Section 3.4 for specifics on the TIU domain and dataset).

Figure 2(a) shows run time for a dataset of n=5 × 10^4 and number of CPUs used to complete one MCMC iteration (not including the initial nearest neighbor set search time, which is common across models). Two versions of the Collapsed model are shown, one assumes the permutation matrix **P** is diagonal (labeled *no perm*) and the other allows CHOLMOD to select an approximately optimal permutation matrix (labeled *perm*). Here, and in other experiments, using a fill-in reducing permutation matrix provides substantial time efficiency gains. The Response model provides full posterior inference on all parameters, with the exception of **w**, and dramatically faster run time compared to the Collapsed model. Inference for the Conjugate model, including $\widehat{\boldsymbol{\beta}}$ and $\widehat{\sigma^2}$ (Algorithm 5), requires about the same amount of time as one Response model MCMC iteration. Explicitly updating **w** is relatively slow;

hence, the Sequential model's computing time falls somewhere between that of the Collapsed and Response models.

For all models, Figure 2(a) show marginal improvement in run time beyond ~6 CPUs and negligible improvement beyond ~12 CPUs. We attribute the slight increase in run time beyond ~12 CPU seen in some models to communication overhead. Run time is actual execution time, or "wall clock time", of the specified number of MCMC iterations. Points of diminishing return on number of CPUs used will change with n; however, exploratory analysis across the range of n considered here suggested 12 CPUs is the bound for substantial gains (clearly this also depends on computing environment and programming decisions).

Figure 2(b) shows time required to execute one sampler iteration by n. The Response and Conjugate models deliver inference across n in ~1/3 and ~1/10 the time required by the Sequential and Collapsed models, respectively. For n=1 × 10^7 the run time is approximately 28, 13, 13, and 95 seconds for the Sequential, Response, Conjugate, and Collapsed, respectively.

3.3 Experiment #2

This experiment compared parameters estimates and predictive performance among the NNGP models for a large dataset. Also, the potential to identify *optimal* values of ϕ and α via cross-validation was assessed for the Conjugate model. We generated observations at 6 × 10^4 locations within a unit square domain from model (2), the $n \times n$ spatial covariance matrix **C** was formed using (1) with ν fixed at 0.5, and the mean comprised an intercept and covariate $\mathbf{x}_1$ drawn from independent $N(0, 1)$. Observations were then generated using the parameter values given in the column labeled *True* in Table 1. Observations at $n = 5 \times 10^4$ of these locations, selected at random, were used to estimate model parameters. Observations at the remaining 1 × 10^4 holdout locations were used to assess model predictive performance.

Following Section 2.3, five-fold cross-validation aimed at minimizing RMSPE and continuous rank probability score (CRPS; Gneiting and Raftery, 2007) for the Conjugate model are given in Figure 3. We observe that a broad range of ϕ and α values deliver comparable predictive performance, and minimization of RMSPE and CRPS yield approximately the same estimates of ϕ and α.

In addition to RMSPE and CRPS, percent of holdout observations covered by their corresponding predictive distribution 95% credible interval (PCI), and mean width of the predictive distributions' 95% credible interval (PIW) were used to assess NNGP model predictive performance. Results given in Table 1 show the NNGP models yield comparable parameter estimates and prediction. Here, the Conjugate model's ϕ and α were selected to minimize RMSPE (results are comparable for minimization of CRPS).

Candidate models' Gelman-Rubin (Gelman and Rubin, 1992) potential scale reduction factor figures and MCMC chain trace plots are given in Figures S2 - S5 of the web supplement. These figures show the Response and Collapsed models provide faster chain convergence for the intercept and spatial covariance parameters compared to Sequential model. Additional analysis in Section S3 of the web supplement reveal that for a smaller dataset generated using the same model, the Sequential model parameter posteriors do not match well that of the full GP.

3.4 Tanana Inventory Unit forest canopy height

Our goal is to create a high-resolution forest canopy height data product, with accompanying uncertainty estimates for prediction and spatial correlation parameters, for the US Forest Service Tanana Inventory Unit (TIU) that covers a large portion of Interior Alaska using a sparse sample of LiDAR data from NASA Goddard's LiDAR, Hyperspectral, and Thermal (G-LiHT) Airborne Imager (Cook et al., 2013).

For remote forested regions, combining sparse airborne LiDAR data with a sparse network of forest inventory data provides a cost-effective means to deliver predictive maps of forest canopy height. In this study, LiDAR data were acquired across the US Forest Service Tanana Inventory Unit (TIU) in Interior Alaska, approximately 140,000 km^2, using the NASA Goddard's LiDAR, Hyperspectral, and Thermal (G-LiHT) Airborne Imager (Cook et al., 2013). The G-LiHT instrument package simultaneously acquires data from a suite of remote sensing instruments to collect complementary information on forest structure (LiDAR), vegetation composition (hyperspectral), and forest health (hyperspectral and thermal).

Here, we consider G-LiHT LiDAR data collected during a 2014 TIU flight campaign. The campaign collected a systematic sample covering ~8% of the TIU, with 78 parallel flight lines spaced ~9 km apart, Figure 4(a), along with incidental measurements to-and-from the transects. The nominal flying altitude of data collection in the TIU was 335 m above ground level, resulting in a sample swath width of ~180 m (30° field of view) and sample density of 3 laser pulses m^2. Point cloud data were classified and used to generate bare earth elevation and canopy height models at 1 m ground sample distance, as described in Cook et al. (2013). G-LiHT point cloud data and derived products are available online at http://gliht.gsfc.nasa.gov. The data was processed following methods in Cook et al. (2013), such that 28,751,400 LiDAR-based estimates of forest canopy height were available on a 15 × 15 m grid along the flight lines. Each grid cell yielded an estimate of canopy height calculated as the height below which 95% of the pulse data was recorded. The subsequent analysis uses a random sample of 5.025×10^6 observations from the larger LiDAR dataset.

Two predictors that completely cover the TIU were considered. First, a Landsat derived percent tree cover data product developed by Hansen et al. (2013), shown as the gray scale surface in Figure 4(a). This product provides percent tree cover estimates for peak growing season in 2010 (most recent year available) and was created using a regression tree model applied to Landsat 7 ETM+ annual composites. These data are provided by the United States Geological Survey (USGS) on an approximate 30 m grid covering the entire globe (Hansen et al., 2013). Second, the perimeters of past fire events from 1947–2014 were obtained from the Alaska Interagency Coordination Center Alaska fire history data product (AICC, 2016). Forest recovery/regrowth following fire is very slow in Interior Alaska. Hence we discretized the fire history data to 1 if the fire occurred within the past 20 years and 0 otherwise, Figure 4(b).

We explored the relationship between canopy height, tree cover and fire history using a non-spatial regression model and NNGP Response, Collapsed, and Conjugate models. We did not consider the Sequential model here because of the convergence issues seen in the preceding experiments. Exploratory analysis using the non-spatial regression suggested both predictors explain a substantial portion of variability in canopy height (Table 2), with a positive association between canopy height and tree cover (TC) and negative association between canopy height and recent fire occurrence (Fire). These results are consistent with our understanding of the TIU forest system. The tree cover variable captures forest canopy sparseness—with sparser canopies resulting in LiDAR height percentiles shifted toward the ground. Recently burned areas are typically replaced with regenerating, shorter stature, forests.

For all models, the intercept and slope regression parameters were given flat prior distributions. The variance components τ^2 and σ^2 were assigned inverse-Gamma $IG(2, 10)$ priors. We assumed an Exponential spatial correlation function with a uniform $U(0.1, 10)$ prior on the decay parameter. The support on the decay corresponds to an effective spatial range between 0.3 to 30 km. Observations at n=5 × 10^6 locations, selected at random, were used to estimate model parameters. Observations at the remaining 2.5 × 10^4 holdout locations were used to assess model predictive performance. Parameter estimates and prediction performance summaries for candidate models are given in Table 2. Results for the m=15 and m=25 models were indistinguishable, hence only m=15 results are presented. Here, NNGP models provide approximately the same predictive performance, and a substantial improvement over the non-spatial regression.

As suggested by Figure 2(b), and seen again here, the Collapsed model using a fill reducing permutation and 12 CPU requires an excessively long run time, i.e., about two weeks to generate 25 × 10^3 MCMC samples. If one is willing to forgo estimates of spatial random effects, the Response model offers greatly improved run time, i.e., about 1.5 days, and parameter and prediction inference comparable to the Collapsed model. The Conjugate model delivers the shortest run time and predictive inference comparable to the other NNGP models.

Figure 4(b) identifies two example areas selected to illustrate how LiDAR and the other data inform forest canopy height prediction. As suggested by the prediction metrics in Table 2, all three NNGP models delivered nearly identical prediction map products. Figure 5 shows the posterior predictive distribution mean and standard deviation from the Response model with m=15 for the two areas. Here, the left subplots identify LiDAR data locations as black points along the flight lines. The presence of strong residual spatial autocorrelation results in fine-scale prediction within, and adjacent to, the flight lines (Figures 5(a)(c)) and more precise posterior predictive distributions as reflected in the standard deviation maps (Figures 5(b)(d)). Predictions more than a km from the flight lines are informed primarily by tree cover and fire occurrence predictors.

The TIU forest's vertical and horizontal structure is highly heterogeneous due, in large part to topography, hydrology, and disturbance history, e.g., fire. This heterogeneity is reflected in the relatively short estimated effective range of just over 1 km (Table 2).

These results provide key input needed for planning future LiDAR campaigns to collect data to inform canopy height models. Using more informative predictor variables would certainly improve prediction across the TIU; however, few complete-coverage high spatial resolution data layers exist, other than those produced using moderate spatial resolution remote sensing products, e.g., the Landsat based tree cover predictor used here.

As seen here, high spatial resolution wall-to-wall map predictions can be achieved with sufficient LiDAR coverage and use of fine-scale residual spatial structure. The G-LiHT LiDAR data—spatially dense along the 180 m swath widths—could better inform canopy height prediction across the TIU if it covered a larger swath width. This could be accomplished by increasing the flight altitude. While a higher nominal flying altitude will increase the swath width, it will also decrease the spatial density of LiDAR observations. Our results suggest that LiDAR density is less important than coverage width, given models were fit using only ~17% (5×10^6/28,751,400) of available data and even then it appears we had ample information to inform prediction within flight lines. This observation has implications for the other LiDAR collection campaigns, e.g., ICESat-2 (Abdalati et al., 2010; ICESat-2, 2015) and Global Ecosystem Dynamics Investigation LiDAR (GEDI, 2014), when they choose between pulse density and swath width.

4 Summary

Our aim has been to propose alternate formulations and derivatives of Bayesian NNGP models developed by Datta et al. (2016a) to substantially improve computational efficiency for fully process-based inference. These improvements make it feasible to bring a rich set of hierarchical spatial Gaussian process models to bear on data intensive analyses such as the TIU forest canopy mapping e ort. Analysis of simulated data shows that compared with the Sequential specification of Datta et al. (2016a), the Response and Collapsed models offer improved MCMC chain behavior for the intercept and spatial covariance parameters. If full inference about the spatial random effects is of interest, then the Response or Conjugate models are not appropriate. So while the Collapsed model can be computationally intensive, depending on the burden imposed by the sparse Cholesky decomposition, it is the only fully Bayesian alternative to the sequential Gibbs sampler developed in Datta et al. (2016a) and should generally be selected over the latter due to its significantly improved chain convergence. Furthermore, recent work by Katzfuss and Guinness (2017) shows that the collapsed model provides a better approximation of the full GP than the Response model in the sense of Kullback-Leibler divergence from the full GP model. If model parameter estimation and/or spatial interpolation of the response is the primary objective, the Response model offers substantial computational gains over the Collapsed model. Finally, relative to the other NNGP models, the Conjugate model delivers massive gains in computational efficiency and seemingly uncompromised predictive inference, but requires specification of the models' spatial decay and α parameters. However, as demonstrated in the simulation and TIU analyses, these parameters can be effectively selected via cross-validation. The response and conjugate NNGP models are available for public use in the spNNGP package (Finley et al., 2017) in R.

The Response model emerges a viable option for obtaining full Bayesian inference about spatial covariance parameters and prediction units. A fully Bayesian kriging model capable of handling 5×10^6 observations on standard computing architectures is an exciting advancement and opens the door to using a rich set of process models to tackle complex problems in big data settings. For example, the Response and Collapsed NNGP models can seamlessly replace Gaussian Processes within multivariate, space-varying coefficients, and space-time settings (see, e.g., Datta et al., 2016a,c,b). The Conjugate model provides a new tool for delivering fast interpolation with few inferential concessions. Extension of the Conjugate model to some of the more complex hierarchical frameworks noted above provides an additional avenue for development.

The TIU analysis shows the advantage of embedding the NNGP as a sparsity-inducing prior within a hierarchical modeling framework. The proposed NNGP specifications yield complete coverage forest canopy height prediction maps with associated uncertainty estimates using sparsely sampled but locally dense $n = 5 \times 10^6$ LiDAR data. The resulting data product is the first statistically robust map of forest canopy for the TIU. Insight into residual spatial dependence will help guide planning for upcoming LiDAR data collection campaigns at global and local scales to improve prediction by leveraging information in more optimally located canopy height observations.

There remains much to be explored in NNGP models. Recent investigations by Guinness (2018) suggest that the Kullback-Leibler divergence between full Gaussian process likelihoods and Vecchia-type nearest neighbor approximations can be sensitive to topological ordering. Our preliminary explorations seem to suggest that while the Kullback-Leibler divergence from the truth may be affected, substantive inference in the form of parameter estimates and predictive performance (based upon root-mean-square-predictions) are very robust. Guinness (2018) also demonstrated empirically that certain carefully chosen orderings of the locations lead to a better approximation of the full GP by NNGP, than what is achieved by the simple co-ordinate based ordering. All the algorithms we propose here are flexible to the choice of ordering. While we have continued to use co-ordinate based ordering for all the data analysis here, we could as easily use any of the orderings proposed by Guinness (2018). We are currently conducting further investigations with the ordering suggested by Guinness (2018) and intend to report on our findings in a subsequent work.

A limiting factor for the hybrid approach adopted in the conjugate NNGP model is the cross-validation procedure for selecting the hyper-parameters. For most spatial applications, the isotropic Matérn functions are often the preferred choice for the covariance kernel, and is convenient for implementing the conjugate model as it only involves two or three unknown parameters. Hence, cross-validation using a grid search on a three dimensional space is computationally feasible. However, as pointed out by one reviewer, many other GP-based applications use more complex covariance functions involving several parameters. For example, in computer model emulations, separable Gaussian covariance functions are commonly used, for which there is a co-ordinate specific range parameter. As with all cross-validation based procedures, the conjugate model will also suffer from the curse of dimensionality in such richly parametrized settings, as searching for optimal or near-optimal points in a high-dimensional space is highly inefficient. Newer strategies need to be conceived for hyper-parameter estimation in such settings.

Another pertinent matter concerns the performance of NNGP models for nonstationary processes. Naive implementations using neighbor selection based on simple Euclidean metrics may not be desirable. Here, the dynamic neighbor-finding algorithms proposed by Datta et al. (2016c) in spatiotemporal contexts may offer a better starting point than finding suitable metrics to choose neighbors. Still, work needs to be done in developing and analyzing analogous algorithms for nonstationary processes. Finally, there is scope to explore NNGP models for high-dimensional multivariate outcomes using spatial factor models (Taylor-Rodriguez et al., 2018) or Graphical Gaussian models and assessing their efficiency for highly complex multivariate spatial datasets.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

Finley was supported by National Science Foundation (NSF) DMS-1513481, EF-1137309, EF-1241874, and EF-1253225. Cook, Morton, and Finley were supported by NASA Carbon Monitoring System grants. Banerjee was supported by NSF DMS-1513654, NSF IIS-1562303 and NIH/NIEHS R01-ES027027.

References

Abdalati W, Zwally H, Bindschadler R, Csatho B, Farrell S, Fricker H, Harding D and Kwok R, Lefsky M, Markus T, Marshak A, Neumann T, Palm S, Schutz B, Smith B, Spinhirne J, and Webb C (2010), "The ICESat-2 Laser Altimetry Mission," Proceedings of the IEEE, 98, 735–751.

AICC (2016), "Fire history in Alaska," http://afsmaps.blm.gov/imf_firehistory/imf.jsp?site=firehistory, accessed: 3–8-16.

Amestoy PR, Davis TA, and Du IS (1996), "An Approximate Minimum Degree Ordering Algorithm," SIAM Journal on Matrix Analysis and Applications, 17, 886–905.

Amestoy PR, Davis TA, and Du IS (2004), "Algorithm 837: AMD, an Approximate Minimum Degree Ordering Algorithm,"ACM Transactions on Mathematical Software, 30, 381–388.

Banerjee S (2017), "High-Dimensional Bayesian Geostatistics," Bayesian Analysis, 12, 583–614. [PubMed: 29391920]

Banerjee S, Gelfand AE, Finley AO, and Sang H (2008), "Gaussian Predictive Process Models for Large Spatial Datasets," Journal of the Royal Statistical Society, Series B, 70, 825–848.

Barbian MH and Assunção RM (2017), "Spatial subsemble estimator for large geostatistical data," Spatial Statistics, 22, 68–88.

Chen Y, Davis TA, Hager WW, and Rajamanickam S (2008), "Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate," ACM Transactions on Mathematical Software, 35, 1–14.

Cook B, Corp L, Nelson R, Middleton E, Morton D, McCorkel J, Masek J, Ranson K, Ly V, and Montesano P (2013), "NASA Goddards LiDAR, Hyperspectral and Thermal (G-LiHT) Airborne Imager," Remote Sensing, 5, 4045–4066.

Cressie N and Johannesson G (2008), "Fixed Rank Kriging for very large spatial data sets," Journal of the Royal Statistical Society B, 70, 209–226.

Dagum L and Menon R (1998), "OpenMP: an industry standard API for shared-memory programming," Computational Science & Engineering, IEEE, 5, 46–55.

Datta A, Banerjee S, Finley AO, and Gelfand AE (2016a), "Hierarchical Nearest-Neighbor Gaussian Process Models for Large Geostatistical Datasets," Journal of the American Statistical Association, 111, 800–812. [PubMed: 29720777]

Datta A, Banerjee S, Finley AO, and Gelfand AE (2016b), "On Nearest-Neighbor Gaussian Process Models For Massive Spatial Data,"Wiley Interdisciplinary Reviews: Computational Statistics, 8, 162–171. [PubMed: 29657666]

Datta A, Banerjee S, Finley AO, Hamm NAS, and Schaap M (2016c), "Nonsep-arable Dynamic Nearest Neighbor Gaussian Process Models for Large Spatio-Temporal Data With an Application to Particulate Matter Analysis," Annals of Applied Statistics, 10, 1286–1316. [PubMed: 29657659]

Davis TA (2006), *Direct Methods for Sparse Linear Systems*, Philadelphia, PA: Society for Industrial and Applied Mathematics.

Davis TA (2016a), "A suite of of sparse matrix software," www.suitesparse.com, accessed 2016–01-01.

Davis TA (2016b), "User Guide for CHOLMOD: a sparse Cholesky factorization and modification package," www.suitesparse.com, accessed 2016–01-01.

Finley A, Datta A, and Banerjee S (2017), spNNGP: Spatial Regression Models for LargeDatasets using Nearest Neighbor Gaussian Processes, r package version 0.1.1.

Finley AO, Sang H, Banerjee S, and Gelfand AE (2009), "Improving the performance of predictive process modeling for large datasets," Computational statistics & data analysis, 53, 2873–2884. [PubMed: 20016667]

Finney MA (2004), "FARSITE: Fire Area Simulatormodel development and evaluation," Tech. Rep. Research Paper RMRS-RP-4, U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station.

Furrer R (2016), spam: SPArse Matrix, r package version 1.4–0.

Furrer R and Sain SR (2010), "spam: A Sparse Matrix R Package with Emphasis onMCMC Methods for Gaussian Markov Random Fields," J. Stat. Softw, 36, 1–25.

GEDI (2014), "Global Ecosystem Dynamics Investigation LiDAR," http://science.nasa.gov/missions/gedi/, accessed: 1–5-2015.

Gelman A and Rubin D (1992), "Inference from iterative simulation using multiple sequences," Statistical Science, 7, 457–511.

Gerber F (2017), gapfill: Fill Missing Values in Satellite Data, r package version 0.9.5.

Gneiting T and Raftery AE (2007), "Strictly Proper Scoring Rules, Prediction, andEstimation," Journal of the American Statistical Association, 102, 359–378.

Gramacy RB (2016), "laGP: Large-Scale Spatial Modeling via Local Approximate Gaussian Processes in R," Journal of Statistical Software, 72, 1–46.

Gramacy RB and Apley DW (2015), "Local Gaussian Process Approximation for Large Computer Experiments," Journal of Computational and Graphical Statistics, 24, 561–578.

Gramacy RB and Sun F (2017), laGP: Local Approximate Gaussian Process Regression, r package version 1.5–1.

Guhaniyogi R and Banerjee S (2018), "Meta-Kriging: Scalable Bayesian Modeling andInference for Massive Spatial Datasets," Technometrics, 0, 1–15.

Guinness J (2018), "Permutation and Grouping Methods for Sharpening Gaussian ProcessApproximations," Technometrics, 0, 1–15.

Hager WW (2002), "Minimizing the Profile of a Symmetric Matrix," SIAM Journal onScientific Computing, 23, 1799–1816.

Hansen MC, Potapov PV, Moore R, Hancher M, Turubanova SA, Tyukavina A, Thau D, Stehman SV, Goetz SJ, Loveland TR, Kommareddy A, Egorov A, Chini L, Justice CO, and Townshend JRG (2013), "High-Resolution Global Maps of 21st-Century Forest Cover Change," Science, 342, 850–853. [PubMed: 24233722]

Heaton MJ, Datta A, Finley AO, Furrer R, Guhaniyogi R, Gerber F, Gramacy RB, Hammerling D, Katzfuss M, Lindgren F, Nychka DW, Sun F, and Zammit-Mangion A (2017), "Methods for

Analyzing Large Spatial Data: A Review and Comparison," ArXiv e-prints, https://arxiv.org/abs/1710.05013.

Huang H and Sun Y (2018), "Hierarchical Low Rank Approximation of Likelihoods for Large Spatial Datasets," Journal of Computational and Graphical Statistics, 27, 110–118.

Hurtt GC, Dubayah R, Drake J, Moorcroft PR, Pacala SW, Blair JB, and Fearon MG (2004), "Beyond Potential Vegetation: Combining Lidar Data and a Height-Structured Model for Carbon Studies," Ecological Applications, 14, 873–883.

ICESat-2 (2015), "Ice, Cloud, and Land Elevation Satellite-2," http://icesat.gsfc.nasa.gov/, accessed: 1–5-2015.

Karypis G and Kumar V (1998), "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs," SIAM Journal on Scientific Computing, 20, 359–392.

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

Katzfuss M and Guinness J (2017), "A general framework for Vecchia approximations of Gaussian processes," https://arxiv.org/pdf/1708.06302.pdf.

Klein T, Randin C, and Korner C (2015), "Water availability predicts forest canopy height at the global scale," Ecology Letters, 18, 1311–1320. [PubMed: 26423470]

Lauritzen SL (1996), Graphical Models, Oxford, United Kingdom: Clarendon Press.

Lefsky MA (2010), "A global forest canopy height map from the Moderate Resolution Imaging Spectroradiometer and the Geoscience Laser Altimeter System," Geophysical Research Letters, 37, l15401.

Liu JS, Wong WH, and Kong A (1994), "Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes," Biometrika, 81, 27–40.

Murphy K (2012), Machine Learning: A probabilistic perspective, Cambridge, MA: TheMIT Press.

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 Computational and Graphical Statistics, 24, 579–599.

Ra SW and Kim JK (1993), "Fast mean-distance-ordered partial codebook search algorithm for image vector quantization," IEEE Transactions on Circuits and Systems II, 40, 576–579.

Rue H, Martino S, Lindgren F, Simpson D, Riebler A, Krainski ET, and Fuglstad G-A (2017), INLA: Bayesian Analysis of Latent Gaussian Models using Integrated Nested Laplace Approximations, r package version 17.06.20.

Sang H, Jun M, and Huang JZ (2011), "Covariance approximation for large multivariate spatial data sets with an application to multiple climate model errors," The Annals of Applied Statistics, 2519–2548.

Stein ML, Chi Z, and Welty LJ (2004), "Approximating Likelihoods for Large SpatialData Sets," Journal of the Royal Statistical Society, Series B, 66, 275–296.

Stratton RD (2006), "Guidance on spatial wildland fire analysis: models, tools, and techniques." Tech. Rep. General Technical Report RMRS-GTR-183, U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station.

Stroud JR, Stein ML, and Lysen S (2017), "Bayesian and Maximum Likelihood Estimation for Gaussian Processes on an Incomplete Lattice," Journal of Computational and Graphical Statistics, 26, 108–120.

Sun Y, Li B, and Genton M (2011), "Geostatistics for large datasets," in Advances And Challenges In Space-time Modelling Of Natural Events, eds. Montero J, Porcu E, and Schlather M, Berlin Heidelberg: Springer-Verlag, pp. 55–77.

Taylor-Rodriguez D, Finley AO, Datta A, Babcock C, Andersen H-E, Cook BD, Morton DC, and Baneerjee S (2018), "Spatial Factor Models for High-Dimensional and Large Spatial Data: An Application in Forest Variable Mapping," ArXiv:1801.02078.

Vecchia AV (1988), "Estimation and Model Identification for Continuous Spatial Processes," Journal of the Royal Statistical Society, Series B, 50, 297–312.

Zammit-Mangion A and Cressie N (2017), "FRK: An R Package for Spatial and Spatio-Temporal Prediction with Large Datasets," arXiv preprint arXiv:1705.08105.

Zhang H (2004), "Inconsistent Estimation and Asymptotically Equal Interpolations in Model-Based Geostatistics," Journal of the American Statistical Association, 99, 250–261.

Zhang X (2016), "An Optimized BLAS Library Based on GotoBLAS2." https://github.com/xianyi/OpenBLAS/, accessed 2015–06-01.

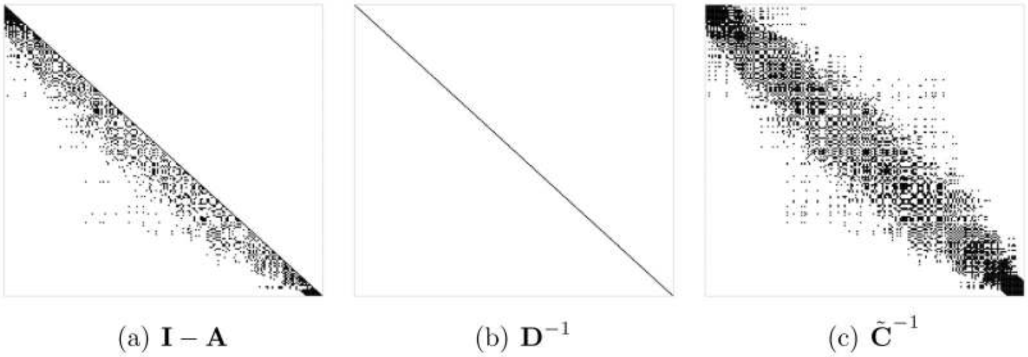


Figure 1:

Structure of the factors making up the sparse $\widetilde{\mathbf{C}}^{-1}$ matrix for n=200 and m=10.

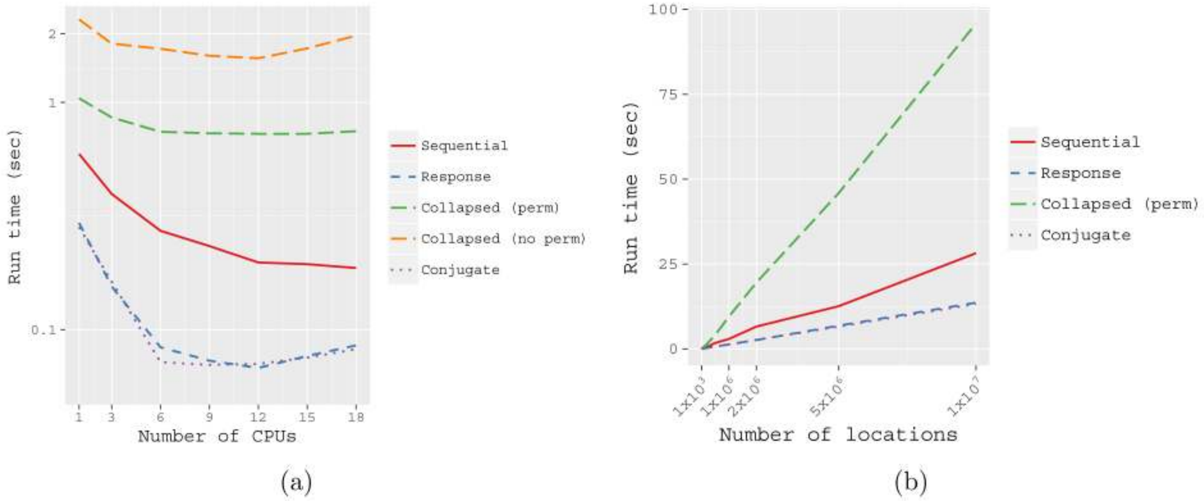
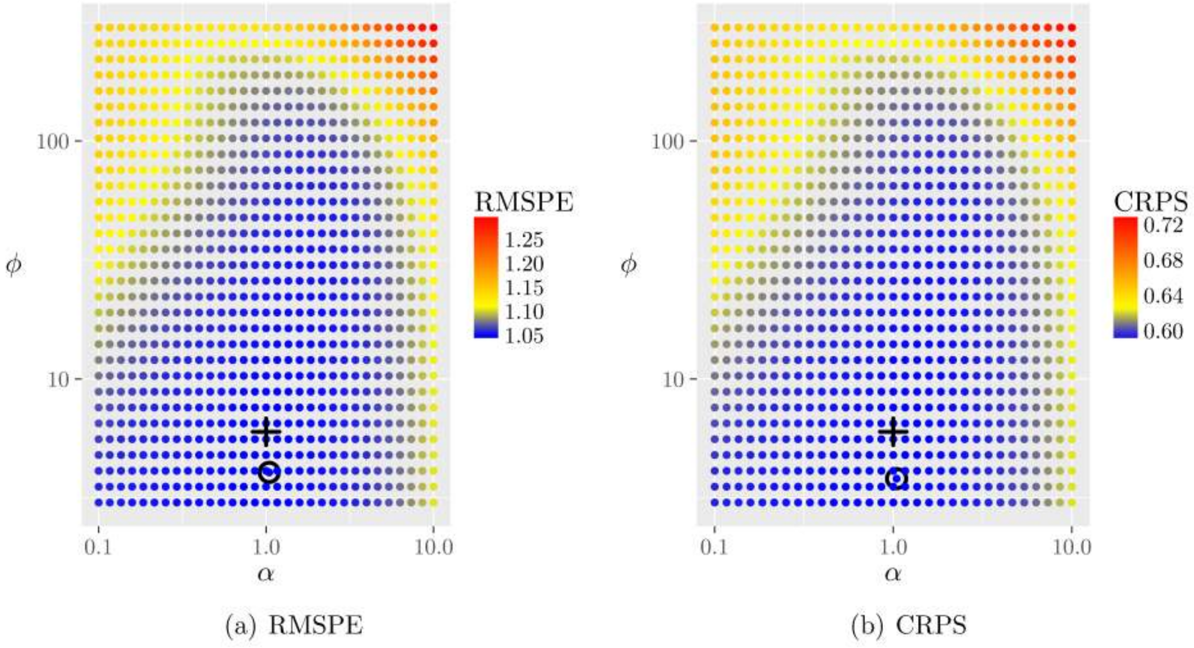


Figure 2:

(a) Run time required for one sampler iteration using n=5 × 10^4 by number of CPUs (y-axis is on the log scale). (b) Run time required for one sampler iteration by number of locations.



(a) RMSPE

(b) CRPS

Figure 3:

Conjugate model cross-validation results for selection of α and ϕ using the simulated dataset. Parameter combination with minimum scoring rule indicated with open circle symbol ○ and true combination used to generate the data indicated with a plus symbol +.

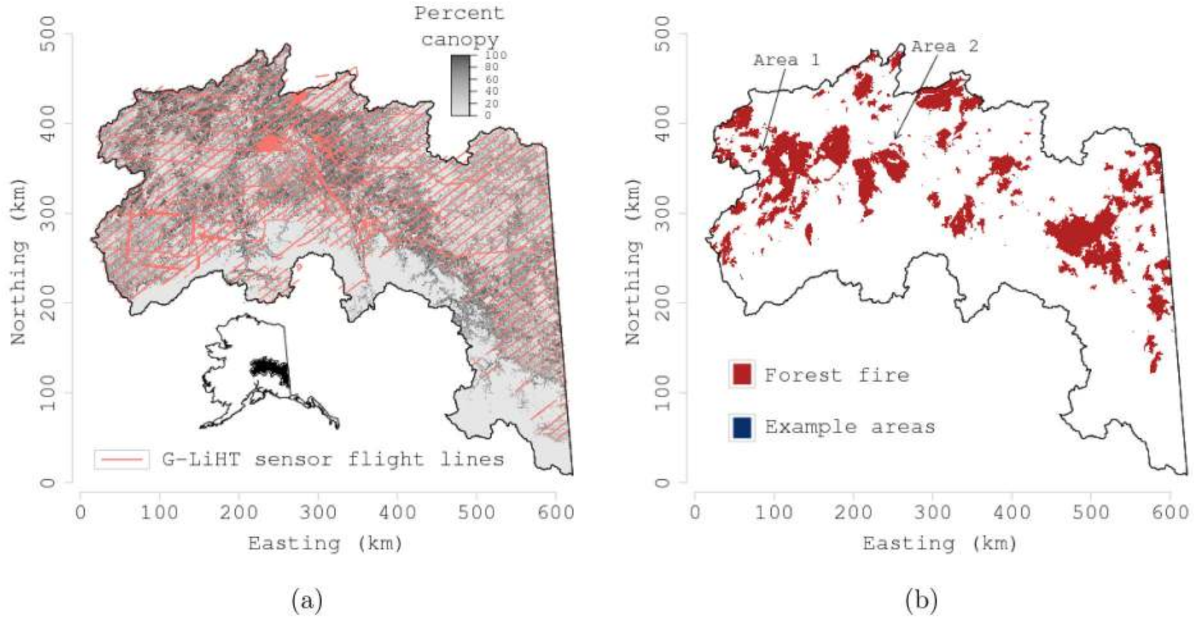
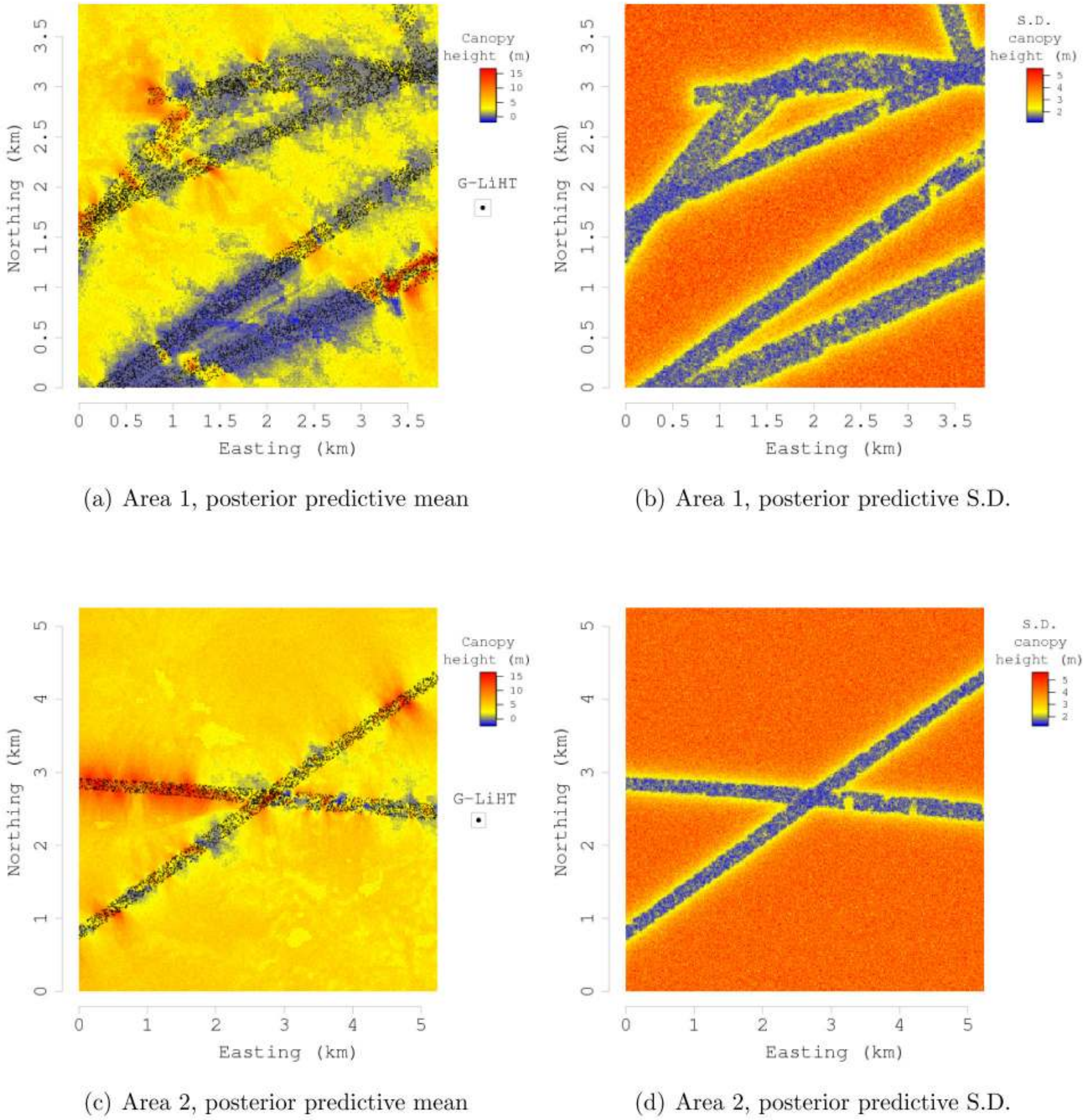


Figure 4:

TIU, Alaska, study region. (a) G-LiHT flight lines where canopy height was measured at 5×10^6 locations and percent tree cover predictor variable. (b) Occurrence of forest fire within the past 20 years predictor variable and two example areas for prediction illustration.



(a) Area 1, posterior predictive mean

(b) Area 1, posterior predictive S.D.

(c) Area 2, posterior predictive mean

(d) Area 2, posterior predictive S.D.

Figure 5:

95th LiDAR percentile height posterior predictive distribution summary at a 30 m pixel resolution for the two example areas identified in Figure 4(b).

Table 1:

Simulated dataset, parameter credible intervals 50% (2.5%, 97.5%) and predictive validation. Bold entries indicate where the true value is not within the 95% credible interval.

Parameter	True	Sequential (metrop)	Sequential (slice)	Response	Collapsed	Conjugate
β_0	1	**0.64 (0.53, 0.75)**	**0.56 (0.44, 0.79)**	**0.84 (0.70, 0.99)**	1.10 (0.51, 1.79)	0.84
β_1	5	5.00 (5.00, 5.01)	5.00 (5.00, 5.01)	5.01 (5.00, 5.01)	5.00 (5.00, 5.01)	5.01
σ^2	1	**1.95 (1.44, 2.21)**	**1.68 (1.11, 2.19)**	1.03 (0.91, 1.21)	**1.69 (1.16, 2.24)**	0.98
τ^2	1	1.00 (0.98, 1.01)	1.00 (0.98, 1.01)	1.00 (0.98, 1.01)	1.00 (0.98, 1.01)	1.02
ϕ	6	**3.39 (3.03, 4.54)**	3.98 (3.04, 6.05)	6.26 (4.88, 7.78)	**3.95 (3.01, 5.83)**	4.05
CRPS		0.59	0.59	0.6	0.59	0.59
RMSPE		1.04	1.04	1.05	1.04	1.05
95% PIC		93.13	92.63	93.08	92.77	94.94
95% PIW		3.87	3.85	3.93	3.84	4.11

Table 2:

TIU dataset results. Parameter credible intervals, 50% (2.5%, 97.5%), predictive validation, and run time for 25×10^3 MCMC iterations.

Parameter	Non-spatial regression	Response	Collapsed	Conjugate minimize RMSPE
β_0	−2.46 (−2.47,−2.45)	2.37 (2.31,2.42)	2.41 (2.35, 2.47)	2.51
β_{TC}	0.13 (0.13, 0.13)	0.02 (0.02, 0.02)	0.02 (0.02, 0.02)	0.02
β_{Fire}	−0.13 (−0.14, −0.12)	0.43 (0.39, 0.48)	0.39 (0.34, 0.43)	0.35
σ^2	–	17.29 (17.13, 17.41)	18.67 (18.50, 18.81)	23.21
τ^2	17.39 (17.37, 17.41)	1.55 (1.54, 1.55)	1.56 (1.55, 1.56)	1.21
ϕ	–	4.15 (4.13, 4.19)	3.73 (3.70, 3.77)	3.83
α	–	–	–	0.052
CRPS	2.3	0.86	0.86	0.84
RMSPE	4.19	1.72	1.73	1.71
95% PIC	93.43	94.29	94.25	94.85
95% PIW	16.27	6.58	6.56	6.73
Run time (hours)	–	38.29	318.81	0.002