Applying Nearest Neighbor Gaussian Processes to Massive Spatial Data Sets: Forest Canopy Height Prediction Across Tanana Valley Alaska

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

May 10, 2017

Abstract

This manuscript addresses the needs for forest scientists to overcome computational hurdles associated with analyzing massive spatial datasets and answering complex inferential questions regarding underlying processes. The primary focus is on reparametrizations and alternate formulations of the recently proposed hierarchical Nearest Neighbor Gaussian Process (NNGP) models (Datta et al., 2016) for improved convergence, better run times, and more robust and reproducible Bayesian inference.

1

Our specific application employs "Light Detection and Ranging" (LiDAR) data to deliver complete coverage forest canopy height prediction maps with associated uncertainty estimates. A major hurdle the very large number of spatial locations (in the order of a few millions). We offer detailed algorithms to ensure efficient CPU memory management and exploit high-performance numerical linear algebra for executing the analysis. Our substantive data analytic contributions pertain to fully process-based posterior inference to accommodate incomplete coverage information from LiDAR instruments, which are essential in advancing our understanding of forest structure and effectively monitoring forest resource dynamics over time. We assess the computational and inferential benefits of these alternate NNGP specifications using simulated data sets and 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

In recent years, the ubiquity of massive geo-referenced datasets in the fields of forestry, ecology, and climate sciences, has resulted in a deluge of statistical methods seeking scalable solutions for massive spatial data analysis. The burgeoning literature on this subject is already too vast to be summarized here and we do not attempt an overview here. Instead, we refer the reader to the review article by Sun et al. (2011), Chapter 12 of Banerjee et al. (2014), and Section 2 of Datta et al. (2016b) for a more recent overview of existing approaches. Current demands for large scale spatial analysis, such as the one we describe below, require statistical inference on underlying spatial processes or random fields at arbitrary resolutions. Bayesian inference is attractive here as it supplies full posterior predictive distributions for the outcomes and the latent processes at arbitrary locations.

As spatial statisticians confront massive datasets with locations $\sim 10^6$ and increasingly demanding inferential questions, several existing approaches that seemed attractive for locations in the order of 10^4 become impractical. Recent methodological developments such as

the Nearest Neighbor Gaussian Process (NNGP: Datta et al., 2016) and multi-resolution approximations (Katzfuss, 2016) propose massively scalable spatial processes. However, to date the focus has primarily rested on theoretical and methodological developments with scant attention to the algorithmic details needed for effectively applying them. For example, Datta et al. (2016) implement a "sequential" Gibbs sampler for updating the latent process that is prone to high autocorrelations and slower convergence. The aforementioned articles do not discuss possible reparametrizations for massively scalable processes and 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. Similar investigations for other competing models, including that of Katzfuss (2016), is beyond the scope of a single manuscript. We outline three alternate formulations of the NNGP models that are significantly more efficient for practical implementation than Datta et al. (2016). Along with the accompanying code supplied with this manuscript, our intended contribution is well aligned with this journal's recent emphasis on reproducible analysis for challenging data analysis in the context of massive spatial datasets.

Our scientific application specifically concerns forest science and, in particular, to create fine resolution canopy height predictions using associated, but misaligned, LiDAR data. Spatially explicit estimates of forest canopy height are key inputs to a variety of ecosystem and Earth system modeling efforts. For example, computer models of forest fire behavior use canopy heights to calculate the probability of crown fire initiation, rate and direction of fire spread via spotting (i.e., ignition from embers leaving the canopy), volume of crown fuels, and localized wind behavior (Finney, 2004; Stratton, 2006). Accurate estimates of forest canopy height also inform dynamic global vegetation models. Forest height can be used as a proxy for forest age or site productivity (Lefsky, 2010; Klein et al., 2015), as such estimates of forest height are important constraints in ecosystem models that estimate forest carbon stocks and fluxes (Hurtt et al., 2004). Variability in forest structure also influences

coupling between the land surface and atmosphere in Earth system models, typically using estimates of surface roughness defined at different length scales (e.g., Sellers et al., 1997). Full Bayesian inference through posterior samples of model parameters and latent processes accurately account for uncertainty estimates in forest canopy height and can be propagated through the computer models to yield more robust error quantification.

Given the scientific and applied interest in forest structure, there is increasing demand for wall-to-wall forest canopy height data at national and biome scales. To date, information about canopy structure has been developed from sparse samples of field data or LiDAR from NASA's Geoscience Laser Altimeter System (GLAS) aboard the ICESat satellite (ICESat, 2016) and ancillary satellite data products (Lefsky, 2010; Simard et al., 2011; Baccini et al., 2004). However, it is difficult to infer fine-scale spatial heterogeneity in canopy structure or forest dynamics from gap-forming disturbances using spatially and temporally limited sampling approaches, respectively. Next generation LiDAR instruments capable of global-scale mapping of forest canopy characteristics, such as ICESat-2 (Abdalati et al., 2010; ICESat-2, 2015) and Global Ecosystem Dynamics Investigation LiDAR (GEDI, 2014), will sample forest features using LiDAR instruments in long transects or cluster designs. Bayesian hierarchical models will be used to create wall-to-wall predictive maps of forest variables from ICESat-2 and GEDI LiDAR samples; knowledge of spatial dependence among LiDAR observations needed to develop such products will also inform future LiDAR sampling designs to maximize some information gain criterion (see, e.g., Xia et al., 2006; Mateu and Müller, 2012). For example, estimates of residual spatial correlation parameters, such as spatial range, are needed to inform decisions about *where* and *at what density* LiDAR samples should be collected.

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

4

Thermal (G-LiHT) Airborne Imager (Cook et al., 2013). Similar efforts to create wall-to-wall canopy height data products from sparse LiDAR data were conducted by Simard et al. (2011) and Fayad et al. (2016). Both studies used data from ICESat-GLAS which, like G-LiHT, collects transects of LiDAR data. Using sparse canopy height estimates from GLAS and several complete coverage topography and climate predictors, Simard et al. (2011) used a Random Forest regression technique (Breiman, 2001) to create a global coverage forest canopy product. Similarly, Fayad et al. (2016) experimented with Random Forest algorithms and kriging to predict forest canopy using GLAS and a set of climate, topography, and other remotely sensed complete coverage predictors. Their results showed that even after using Random Forests, residual kriging improved predictions. This result is not surprising because forest canopy structure typically exhibits strong spatial correlation due to forest response to disturbances acting across spatial scales (e.g., timber harvesting, fire, insects, and wind) and site characteristics (e.g., soil properties, topography, and climate). Indeed, results in Section 3.4 show simple linear models that use available disturbance and site quality predictors is inadequate to explain forest canopy height variability across the TIU.

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 important and efficient reparametrizations: the collapsed NNGP model, an NNGP model for the outcomes themselves (with no latent process), and a conjugate NNGP model that allows MCMC-free inference. Section 3 offers several detailed simulation experiments on model performance and assessment and also presents a detailed analysis of the 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 $y(\mathbf{s}_i) = \mathbf{x}(\mathbf{s}_i)^\top \boldsymbol{\beta} + w(\mathbf{s}_i) + \epsilon(\mathbf{s}_i)$, 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 com-

monly 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 $c_{ij} = C(\mathbf{s}_i, \mathbf{s}_j \,|\, \boldsymbol{\theta})$ and $\boldsymbol{\theta}$ denotes the GP covariance parameters. A popular choice for $C(\cdot, \cdot \,|\, \boldsymbol{\theta})$ is the Matérn covariance function specified as:

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

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

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

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, implementation of model (2) poses multiple computational roadblocks. Firstly, storing the matrix $\mathbf{C}$ requires $O(n^2)$ dynamic memory. Furthermore, evaluating $N(\mathbf{w} \,|\, \mathbf{0}, \mathbf{C})$ 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 $\sim 10^4$–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 Nearest Neighbor Gaussian Processes (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 $p(\mathbf{w}) = p(w_1) \prod_{i=2}^n p(w_i \,|\, \mathrm{Pa}[i])$, where $w_i \equiv w(\mathbf{s}_i)$ and $\mathrm{Pa}[i] = \{w_1, w_2, \ldots, w_{i-1}\}$ is the set of parents of w_i. We can construct sparse models for $\mathbf{w}$ by shrinking the size of $\mathrm{Pa}[i]$. In spatial contexts, this can be done by defining $\mathrm{Pa}[i]$ 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 have been studied and exploited by Vecchia (1988); Stein et al. (2004); Stroud et al. (2014);

6

Datta et al. (2016,c). The NNGP builds upon the above 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 $N(\mathbf{w} \,|\, \mathbf{0}, \mathbf{C})$ as a linear model,

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

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 diagonal entries $d_{11} = \mathrm{var}(w_1)$ and $d_{ii} = \mathrm{Var}(w_i \,|\, \{w_j : j < i\})$ for $i = 2, \ldots, n$.

From the structure of $\mathbf{A}$ it is evident that $\mathbf{I} - \mathbf{A}$ is nonsingular and $\mathbf{C} = (\mathbf{I} - \mathbf{A})^{-1} \mathbf{D} (\mathbf{I} - \mathbf{A})^{-\top}$. For any matrix $\mathbf{M}$ and set of indices $I_1, I_2 \subseteq \{1, 2, \ldots, n\}$, let $\texttt{M}[I_1, I_2]$ denote the submatrix of $\mathbf{M}$ formed by the rows indexed by I_1 and columns index by I_2. Note that $\texttt{D[1,1]} = \texttt{C[1,1]}$ and the first row of $\mathbf{A}$ is $\mathbf{0}$. A pseudocode to compute the remaining elements of $\mathbf{A}$ and $\mathbf{D}$ is

$$
\begin{aligned}
&\texttt{for(i in 1:(n-1)) \{} \\
&\qquad \texttt{A[i+1,1:i] = solve(C[1:i,1:i], C[1:i,i+1])} \\
&\qquad \texttt{D[i+1,i+1] = C[i+1,i+1] - dot(C[i+1,1:i],A[i+1,1:i])} \\
&\qquad \texttt{\},}
\end{aligned}
\tag{3}
$$

where $\texttt{1:i}$ denotes the set $\{1, 2, \ldots, i\}$, $\texttt{solve(B,b)}$ computes the solution $\mathbf{x}$ for the linear system $\mathbf{B}\mathbf{x} = \mathbf{b}$, and $\texttt{dot(u,v)}$ denotes the inner-product between two vectors $\mathbf{u}$ and $\mathbf{v}$.

The above pseudocode computes the Cholesky decomposition of $\mathbf{C}$. There is, however, no apparent gain to be had from the preceding computations since, as the loop runs into higher values of $\texttt{i}$ closer to $\texttt{n}$, the dimension of $\texttt{C[1:i,1:i]}$ 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 $\mathbf{A}$ to be zero. For example, suppose we permit no more than $\texttt{m}$ elements in each row of $\mathbf{A}$ to be nonzero. Let $\texttt{N[i]}$ be the set of indices $\texttt{j} < \texttt{i}$ such that

`A[i,j]` $\neq 0$. One can then compute the elements of $\mathbf{A}$ and $\mathbf{D}$ as:

```
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]])
}.
```

(4)

In (4) we solve `n-1` linear systems of size at most $\mathtt{m} \times \mathtt{m}$ where $m = \max_i |\mathtt{N(i)}|$. 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 $\mathbf{C}$, we construct a sparse strictly lower-triangular matrix $\mathbf{A}$ with no more than $m(\ll n)$ non-zero entries in each row, and the diagonal matrix $\mathbf{D}$ using the pseudocode in (4) 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.
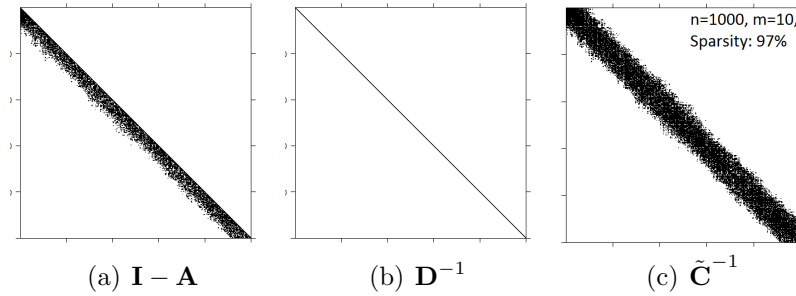

(a) $\mathbf{I} - \mathbf{A}$ (b) $\mathbf{D}^{-1}$ (c) $\tilde{\mathbf{C}}^{-1}$

Figure 1: Structure of the factors making up the sparse $\widetilde{\mathbf{C}}^{-1}$ matrix.

The factorization of $\widetilde{\mathbf{C}}^{-1}$ facilitates cheap computation of quadratic forms $\mathbf{u}^{\top}\widetilde{\mathbf{C}}^{-1}\mathbf{v}$ in terms $\mathbf{A}$ and $\mathbf{D}$. The algorithm to evaluate such quadratic forms `qf(u,v,A,D)` is provided

8

in the following pseudocode:

```
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)]))        (5)

        *(v[i] - dot(A[i,N(i)], v[N(i)]))/D[i,i]

    },
```

where $*$ and $/$ denote multiplication and division by scalars, respectively. Observe (5) 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 a quadratic forms involving a $n \times n$ dense matrix. Also, importantly, the determinant of $\widetilde{\mathbf{C}}$ is obtained with almost no additional cost: it is simply $\prod_{i=1}^{n} \texttt{D[i,i]}$.

Hence, while $\widetilde{\mathbf{C}}$ need not be sparse, the density $N(\mathbf{w} \,|\, 0, \widetilde{\mathbf{C}})$ is cheap to compute requiring only $O(n)$ flops. This was exploited by Datta et al. (2016) where the neighbor sets were constructed based on m nearest neighbors and the traditional GP prior for $\mathbf{w}$ in (2) was replaced with an NNGP prior $N(\mathbf{w} \,|\, \mathbf{0}, \widetilde{\mathbf{C}})$. The Markov chain Monte Carlo (MCMC) implementation of the NNGP model in Datta et al. (2016) requires updating the n latent spatial effects $\mathbf{w}$ 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. In this manuscript, we explore alternate 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

9

merits and demerits are assessed both in terms of computational burden as well as model prediction and inference. Simulation experiments using spatial datasets of upto 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 $N(\mathbf{y} \,|\, \mathbf{X}\boldsymbol{\beta} + \mathbf{w}, \tau^2\mathbf{I}) \, N(\mathbf{w} \,|\, \mathbf{0}, \widetilde{\mathbf{C}})$ and circumvent the sampling of $\mathbf{w}$ in the Gibbs' sampler by marginalizing out $\mathbf{w}$ to have the collapsed NNGP model

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Lambda}) \text{ where } \boldsymbol{\Lambda} = \widetilde{\mathbf{C}} + \tau^2\mathbf{I} \tag{6}$$

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 $u \,|\, \cdot$ 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 $\boldsymbol{\Lambda}^{-1} = \tau^{-2}\mathbf{I} - \tau^{-4}\boldsymbol{\Omega}^{-1}$ where $\boldsymbol{\Omega} = (\widetilde{\mathbf{C}}^{-1} + \tau^{-2}\mathbf{I})$ 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 $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\}$. 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 $\mathtt{dot}(\cdot, \cdot)$ introduced earlier, we require a fill-reducing permutation matrix and a sparse Cholesky factorization ($\mathtt{sparsechol}(\cdot)$) for a sparse positive-definite matrix. Large matrix-matrix and matrix-vector multiplications either involve at least one triangular matrix ($\mathtt{trmm}(\cdot,)$ or $\mathtt{trmv}(\cdot, \cdot)$) or at least one sparse matrix ($\mathtt{sparsemm}(\cdot, \cdot)$ or $\mathtt{sparsemv}(\cdot, \cdot)$). We also use $\mathtt{diagsolve}(\cdot, \cdot)$ and $\mathtt{trsolve}(\cdot, \cdot)$ 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 $\mathtt{chol}(\cdot)$, $\mathtt{gemv}(\cdot, \cdot)$ and $\mathtt{solve}(\cdot, \cdot)$, 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 $\mathtt{diag}(\cdot)$ to extract the diagonal elements of a matrix, $\mathtt{prod}(\cdot)$ to compute the product of the elements in a vector and $\mathtt{rnorm}(\cdot)$ 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 $\boldsymbol{\beta}$:**
   $\boldsymbol{\beta} \mid \cdot \sim N(\mathbf{B}^{-1}\mathbf{b}, \mathbf{B}^{-1})$, 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 (3) to obtain $\mathtt{A}$ and $\mathtt{D}$ using $\mathtt{C}$ and $\{N(i) \mid i = 1, 2, \ldots, n\}$ $O(nm^3)$ flops

     (b) $\Omega = \mathtt{trmm}((\mathtt{I} - \mathtt{A})^\top, \mathtt{diagsolve}(\mathtt{D}, \mathtt{I} - \mathtt{A})) + \tau^{-2} * \mathtt{I}$ $O(nm^2)$ flops

     (c) Find a fill reducing permutation matrix $\mathtt{P}$ for Ω

     (d) $\mathtt{L} = \mathtt{sparsechol}(\mathtt{sparsemm}(\mathtt{sparsemm}(\mathtt{P}, \Omega), \mathtt{P}^\top))$

     (e)   `for (j in 1:n) {`

             $\mathtt{u}_j = \mathtt{trsolve}(\mathtt{L}, \mathtt{sparsemv}(\mathtt{P}, \mathtt{X[,j]}))$; $\mathtt{v}_j = \mathtt{trsolve}(\mathtt{L}^\top, \mathtt{u}_j)$

            `}`

     (f) $\mathtt{F} = \mathtt{solve}(\mathtt{V}_\beta, \mathtt{I})$; $\mathtt{f} = \mathtt{solve}(\mathtt{V}_\beta, \boldsymbol{\mu}_\beta)$ $O(p^3)$ flops

     (g) Solve for $p \times p$ matrix $\mathbf{B}$ and $p \times 1$ vector $\mathbf{b}$: $O(np^2)$ flops

         `for (j in 1:p) {`

               $\mathtt{b[j]} = \mathtt{dot}(\mathtt{y}, \mathtt{X[,j]})/\tau^2 - \mathtt{dot}(\mathtt{y}, \mathtt{sparsemv}(\mathtt{P}, \mathtt{v}_j))/\tau^4 + \mathtt{f[j]}$

               `for (i in 1:p) {`

$$\texttt{B[i,j]} = \texttt{dot(X[,i],X[,j])}/\tau^2 - \texttt{dot(X[,i],sparsemv(P,}\mathbf{v}_j\texttt{))}/\tau^4 + \texttt{F[i,j]}$$

```
        }
    }
```

   (h) $\boldsymbol{\beta} = \texttt{solve(B,b)}$ + $\texttt{trsolve(chol(B),rnorm(p))}$ $O(p^3)$ flops

2: **Metropolis-Hastings (MH) update for $\{\boldsymbol{\theta}, \tau^2\}$:**

$$p(\boldsymbol{\theta}, \tau^2 \,|\, \cdot) \propto p(\boldsymbol{\theta}, \tau^2) \times \frac{1}{\sqrt{\det(\boldsymbol{\Lambda})}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top \boldsymbol{\Lambda}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right)$$

   (a) $\texttt{r} = \texttt{y} - \texttt{gemv(X,}\boldsymbol{\beta}\texttt{)}$; $\texttt{u} = \texttt{trsolve(L, sparsemv(P,r))}$; $\texttt{v} = \texttt{trsolve(L}^\top\texttt{,u)}$ $O(np)$ flops

   (b) $\texttt{q} = \texttt{dot(r,r)}/\tau^2 - \texttt{dot(r,sparsemv(P,v))}/\tau^4$

   (c) $\texttt{d} = \tau^{2*n} * \texttt{prod(diag(D))} * \texttt{prod(diag(L))}^2$ $O(n)$ flops

   (d) Generate $\texttt{p}(\boldsymbol{\theta}, \tau^2 \,|\, \cdot) \propto \dfrac{\texttt{exp(-q/2)} * \texttt{p}(\boldsymbol{\theta}, \tau^2)}{\texttt{sqrt(d)}}$

3: Repeat Steps (1) and (2) (except Step 1(c)) N times to obtain N MCMC samples for $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\}$

Observe that the entire Algorithm 1 is devoid of any expensive operations like $\texttt{solve}$, $\texttt{chol}$ or $\texttt{gemv}$ on dense $n \times n$ matrices. All such operations are limited to $m \times m$ or $p \times p$ matrices. 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 $\mathbf{L}$ in Algorithm 1 (Steps 1(c)-(e)) depends on the data design. Although $\boldsymbol{\Omega}$ is sparse $O(nm^2)$ non-zero entries, the sparsity of its Cholesky factor $\mathbf{L}$ actually depends on the location of the non-zero entries. Hence we used a fill reducing permutation $\mathbf{P}$ that increases the sparsity of the Cholesky factor. Although $\mathbf{P}$ needs to be evaluated only once before the MCMC, finding the optimal $\mathbf{P}$ yielding the least fill-in is a 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 is 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 computations gains can accrue by truncating the MCMC run short. Furthermore, all the $\texttt{for}$ 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 $\mathbf{L}$ and $\mathbf{u}$ 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 $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\}$

1: **Sample from $p(\mathbf{w} \,|\, \cdot)$ one-for-one for each post-convergence sample of $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\}$**
   $\mathbf{w} \,|\, \cdot \sim N(\mathbf{B}^{-1}\mathbf{b}, \mathbf{B}^{-1})$, where $\mathbf{B} = \widetilde{\mathbf{C}}^{-1} + \tau^{-2}\mathbf{I}$ and $\mathbf{b} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})/\tau^2$

   (a) $\mathtt{z} = \mathtt{rnorm}(n)$ $O(n)$ flops

   (b) $\mathtt{w} = \mathtt{sparsemv}(\mathtt{P}^\top, \mathtt{trsolve}(\mathtt{L}^\top, \mathtt{u}/\tau^2 + \mathtt{z}))$

2: **Prediction at a new location $\mathbf{s}_0$:**
   $y(\mathbf{s}_0) \,|\, \cdot \sim N(\mathbf{x}(\mathbf{s}_0)^\top \boldsymbol{\beta} + w(\mathbf{s}_0), \tau^2)$

   (a) Find $\mathtt{N}_0$ — set of m nearest neighbors of $\mathbf{s}_0$ among $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$ $O(n)$ flops

   (b) $\mathtt{c} = \mathtt{C}(\mathbf{s}_0, \mathtt{N}_0; \boldsymbol{\theta})$ $O(m)$ flops

   (c) $\mathtt{m} = \mathtt{dot}(\mathtt{c}, \mathtt{solve}(\mathtt{C}(\mathtt{N}_0, \mathtt{N}_0), \mathtt{w}[\mathtt{N}_0]))$ $O(m^3)$ flops

       $\mathtt{v} = \mathtt{C}(\mathbf{s}_0, \mathbf{s}_0; \boldsymbol{\theta}) - \mathtt{dot}(\mathtt{c}, \mathtt{solve}(\mathtt{C}(\mathtt{N}_0, \mathtt{N}_0), \mathtt{c}))$

   (d) $\mathtt{w}(\mathbf{s}_0) = \mathtt{m} + \mathtt{sqrt}(\mathtt{v}) * \mathtt{rnorm}(1)$ $O(p)$ flops

       $\mathtt{y}(\mathbf{s}_0) = \mathtt{dot}(\mathtt{x}(\mathbf{s}_0), \boldsymbol{\beta}) + \mathtt{w}(\mathbf{s}_0) + \tau * \mathtt{rnorm}(1)$ $O(p)$ 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 $p(\mathbf{w} \,|\, \mathbf{y})$ for the observed locations, use them to sample from $p(w(\mathbf{s}_0) \,|\, \mathbf{y})$ and then from $p(y(\mathbf{s}_0) \,|\, \mathbf{y})$.

2.2 NNGP for the response

Both the sequential NNGP Algorithm in Datta et al. (2016) 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. Although recovering $\mathbf{w}$, 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 $\mathbf{y}$ that sacrifices the ability to recover $\mathbf{w}$ and directly predicts the response at new locations.

Datta et al. (2016) demonstrated that an NNGP model can be derived from any legitimate Gaussian Process. Observe that 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 $\mathbf{y}$, this translates to simply replacing the covariance matrix $\boldsymbol{\Sigma} = \mathbf{C} + \tau^2 \mathbf{I}$ with its nearest-neighbor approximation $\widetilde{\boldsymbol{\Sigma}}$ as described in Section 2. The sparsity properties documented in Section 2 apply to $\widetilde{\boldsymbol{\Sigma}}$ 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 $\boldsymbol{\beta}$:**
$\boldsymbol{\beta} \,|\, \cdot \sim N(\mathbf{B}^{-1}\mathbf{b}, \mathbf{B}^{-1})$, 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 (3) to obtain A and D using Σ and $\{N(i) \,|\, i = 1, 2, \ldots, n\}$ $O(nm^3)$ flops

    (b) $\mathtt{F = solve(V_\beta, I)}$; $\mathtt{f = solve(V_\beta, \boldsymbol{\mu}_\beta)}$ $O(p^3)$ flops

    (c) Solve for $p \times p$ matrix $\mathbf{B}$ and $p \times 1$ vector $\mathbf{b}$ using (5): $O(nmp^2)$ 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) $\boldsymbol{\beta} = \mathtt{solve(B,b)} + \mathtt{trsolve(chol(B), rnorm(p))}$ $O(p^3)$ flops

2: **Metropolis-Hastings (MH) update for $\{\boldsymbol{\theta}, \tau^2\}$:**
$p(\boldsymbol{\theta}, \tau^2 \,|\, \cdot) \propto p(\boldsymbol{\theta}, \tau^2) \times \dfrac{1}{\sqrt{\det(\widetilde{\boldsymbol{\Sigma}})}} \exp\left(-\tfrac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top \widetilde{\boldsymbol{\Sigma}}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right)$

    (a) $\mathtt{e = y - gemv(X, \boldsymbol{\beta})}$;   Using (5), $\mathtt{q = qf(e,e,A,D)}$ $O(n(p+m))$ flops

    (b) $\mathtt{d = prod(diag(D))}$ $O(n)$ flops

    (c) Generate $\mathtt{p(\boldsymbol{\theta}, \tau^2 \,|\, \cdot)} \propto \dfrac{\mathtt{exp(-q/2) * p(\boldsymbol{\theta}, \tau^2)}}{\mathtt{sqrt(d)}}$ $O(1)$ 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 computation 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 $\mathbf{w}$ 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 $\mathbf{w}$ is not possible as highlighted in Datta et al. (2016). 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 $y(\mathbf{s})$ at any arbitrary location $\mathbf{s}$. This is achieved easily through Algorithm 4 given below, where $\mathbf{y}_{N(\mathbf{s}_0)}$ represents the subvector of $\mathbf{y}$ corresponding to the points in $N(\mathbf{s}_0)$, $\mathbf{X}_{N(\mathbf{s}_0)}$ is the corresponding design matrix, and $\boldsymbol{\Sigma}_0$ 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 $p(y(\mathbf{s}_0) \,|\, \cdot)$ one-for-one for each post-convergence sample of $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2\}$**
$$y(\mathbf{s}_0) \,|\, \cdot \sim N(\mathbf{x}(\mathbf{s}_0)^\top \boldsymbol{\beta} + \mathbf{c}_0^\top \boldsymbol{\Sigma}_0^{-1}(\mathbf{y}_{N(\mathbf{s}_0)} - \mathbf{X}_{N(\mathbf{s}_0)} \boldsymbol{\beta}), \Sigma(\mathbf{s}_0, \mathbf{s}_0) - \mathbf{c}_0^\top \boldsymbol{\Sigma}_0^{-1} \mathbf{c}_0)$$

    (a) Find $\mathtt{N_0}$ — set of m nearest neighbors of $\mathbf{s}_0$ among $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$ $O(n)$ flops

    (b) $\mathtt{c} = \Sigma(\mathbf{s}_0, \mathtt{N_0}; \boldsymbol{\theta})$ $O(m)$ flops

    (c) $\mathtt{m} = \mathtt{dot}(\mathtt{c}, \mathtt{solve}(\Sigma[\mathtt{N_0}, \mathtt{N_0}], \mathtt{y}[\mathtt{N_0}] - \mathtt{dot}(\mathtt{X}[\mathtt{N_0},], \boldsymbol{\beta}))$ $O(m^3)$ flops

        $\mathtt{v} = \Sigma(\mathbf{s}_0, \mathbf{s}_0) - \mathtt{dot}(\mathtt{c}, \mathtt{solve}(\Sigma[\mathtt{N_0}, \mathtt{N_0}], \mathtt{c}))$

    (d) $\mathtt{y}(\mathbf{s}_0) = \mathtt{dot}(\mathtt{x}(\mathbf{s}_0), \boldsymbol{\beta}) + \mathtt{m} + \mathtt{sqrt}(\mathtt{v}) * \mathtt{rnorm(1)}$ $O(p)$ flops

---

2.3   MCMC-free exact Bayesian inference using conjugate NNGP

The fully Bayesian approaches developed in Datta et al. (2016) 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 legitimate 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 $N(\mathbf{y} \,|\, \mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{M})$, where $\mathbf{M} = \mathbf{G} + \alpha \mathbf{I}$ and $\mathbf{G}$ denotes the Matern correlation matrix corresponding to the co-

variance 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 and identifiability issues. This has also been established theoretically in an asymptotic setting in Zhang (2004). 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 $IG(\sigma^2 \,|\, a_\sigma, b_\sigma) \times N(\boldsymbol{\beta} \,|\, \boldsymbol{\mu}_\beta, \sigma^2 \mathbf{V}_\beta) \times N(\mathbf{y} \,|\, \mathbf{X}\boldsymbol{\beta}, \sigma^2 \widetilde{\mathbf{M}})$ with joint posterior distribution

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

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 $\sigma^2 \sim IG(a_\sigma^*, b_\sigma^*)$ 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 $\boldsymbol{\beta} \,|\, \mathbf{y} \sim \text{MVS-}t_{2a_\sigma^*}\left(\mathbf{B}^{-1}\mathbf{b}, \frac{b_\sigma^*}{a_\sigma^*} \mathbf{B}^{-1}\right)$ and $\sigma^2 \,|\, \mathbf{y} \sim IG(a_\sigma^*, b_\sigma^*)$, 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 $b_\sigma^*/(a_\sigma^* - 1)$ and $b_\sigma^{*2}/(a_\sigma^* - 1)^2(a_\sigma^* - 2)$, 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. A pseudocode for fast evaluation of the above are provided in Algorithm 5. The marginal posterior predictive distribution at a new location $\mathbf{s}_0$ is given by $y(\mathbf{s}_0) \,|\, \mathbf{y} \sim t_{2a_\sigma^*}(m_0, b_\sigma^* v_0 / a_\sigma^*)$ where expressions for m_0 and v_0 are provided in Step 3 of Algorithm 5. We deploy hyperparameter tuning based on K-fold cross-validation to choose the optimal α and ϕ from a

grid of possible values. 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 kth 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 α and ϕ, 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 σ^2 after removing the kth fold of the data:

   (a) Use (3) to obtain `A(k)` and `D(k)` from $\mathbf{M}[S(-k), S(-k)]$ and $\mathcal{N}$ $O(nm^3)$ flops

   (b) `F = solve(`V_β`, I)` ;   `f = solve(`$V_\beta, \boldsymbol{\mu}_\beta$`)` $O(p^3)$ flops

   (c) Solve for $p \times p$ matrix $\mathbf{B}(k)$ and $p \times 1$ vector $\mathbf{b}(k)$ using (5): $O(nmp^2)$ 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) `V(k) = solve(`$B(k)$`, I)` ; `g(`k`) = gemv(V(k),b(k))` $O(p^3)$ flops

       `a`$_\sigma^*$`(k) = a`$_\sigma$` + (n-n/K)/2`

       `b`$_\sigma^*$`(k) = b`$_\sigma$` + (dot(`$\boldsymbol{\mu}_\beta$`,f) + qf(y[S(-k)],y[S(-k)],A(k),D(k)) - dot(g(k),b(k)))/2`

   (e) $\widehat{\boldsymbol{\beta}}$` = g(k)` ; $\widehat{\sigma^2}$` = b`$_\sigma^*$`(k)/(a`$_\sigma^*$`(k)-1)`

3: Predicting posterior means of $\mathbf{y}[S(k))]$: $O(nm^3/K)$ flops

```
for (s in S(k)) {
    N(s,k) = m-nearest neighbors of s from S(-k)
    z = M(s,N(s,k))
    w = solve(M[N(s,k),N(s,k)],z)
    m0 = ŷ(s) = dot(x(s), g(k)) + dot(w,(y[N(s,k)] - dot(X[N(s,k),],g(k))))
    u = x(s) - dot(X[N(s,k),],w)
    v0 = dot(u, gemv(V(k),u)) + 1 + α - dot(w,z)
    Var(ŷ(s)) = b*_σ(k)v0/(a*_σ(k) - 1)
}
```

4: Root Mean Square Predictive Error (RMSPE) over K folds: $\hspace{2cm} O(n)$ flops

    (a) Initialize $e = 0$

```
for (k in 1:K) for (s_i in S[k]) {
    e= e +(y(s_i)-ŷ(s_i) )²
}
```

5: Cross validation for choosing α and ϕ
    (a) Repeat steps (2) and (3) for G values of α and ϕ $\hspace{1cm} O(GKnm(p^2 + m^2))$ flops
    (b) Choose α_0 and ϕ_0 as the value that minimizes the average RMSPE $\hspace{0.5cm} O(G)$ flops

Parameter estimation and prediction

6: Repeat step (2) with $(\alpha_0, \phi_0)^\top$ and the full data to get $(\boldsymbol{\beta}, \sigma^2)\,|\,\mathbf{y}$ $\hspace{0.5cm} O(nmp^2 + nm^3)$ flops
7: Repeat step (3) with $(\alpha_0, \phi_0)^\top$ and the full data to predict at a new location $\mathbf{s}_0$ to obtain the mean and variance of $y(\mathbf{s}_0)\,|\,\mathbf{y}$ $\hspace{3cm} O(m^3)$ flops

Algorithm 5 completely circumvents MCMC based iterative sampling and only requires at most $O(n)$ flops per step. Although the computationally calculations need to be replicated for every (ϕ, α) combination, unlike the MCMC based algorithms that run serially, this step is embarrassingly 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) space. 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 each analysis, we consider four candidate models labeled: *Sequential* defined in Datta et al. (2016); *Collapsed* defined in Section 2.1; *Response* defined in Section 2.2, and; *Conjugate* defined in Section 2.3. Additionally, analysis of a small simulated dataset is provided in Section S3 of the web supplement to compare full GP and NNGP model parameter estimates.

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`) able to use 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 quadratic function (5) 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 $\mathbf{P}(\widetilde{\mathbf{C}}^{-1} + \tau^{-2}\mathbf{I})\mathbf{P}^\top$, and; solvers for sparse triangular systems.

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 $\mathbf{P}$. 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. (2016), all models were fit using $m=15$ neighbors unless noted otherwise.

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. Prior distributions on regression coefficients β_0 and β_1 were assumed to be *flat*, variance parameters σ^2 and τ^2 followed an inverse-gamma, and spatial decay parameter ϕ followed a uniform with hyper-parameters selected accordingly for the TIU dataset (see Section 3.4).

Figure 2(a) shows run time for a dataset of $n=5 \times 10^4$ 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 $\mathbf{P}$ 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 $\mathbf{w}$, and dramatically faster run time compared to the Collapsed model. Obtaining the Conjugate model $\boldsymbol{\beta}$ and σ^2 point estimates requires about the same amount of time as one Response model MCMC iteration. Explicitly updating $\mathbf{w}$ is relatively slow; hence, the Sequential model's computing time falls somewhere between that of the Collapsed and Response models.
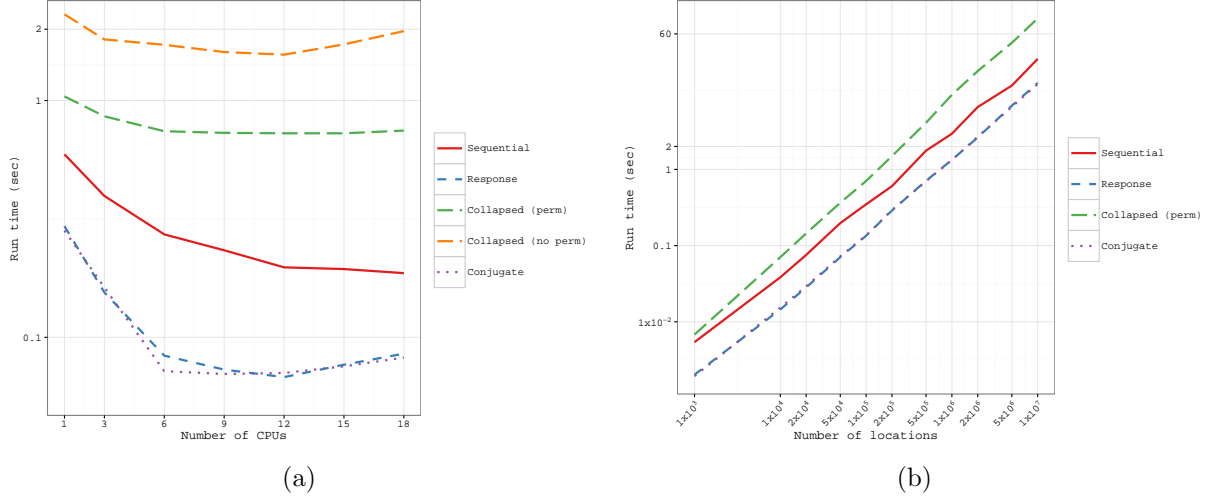
Figure 2: (a) Run time required for one sampler iteration using $n=5 \times 10^4$ by number of CPUs (y-axis is on the log scale). (b) Run time required for one sampler iteration by number of locations (both axes are on the log scale).

For all models, Figure 2(a) show marginal improvement in run time beyond $\sim$6 CPUs and negligible improvement beyond $\sim$12 CPUs. 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 $\sim$1/3 and $\sim$1/10 the time required by the Sequential and Collapsed models, respectively. For $n=1 \times 10^7$ 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 $\mathbf{C}$ was formed using (1) with ν fixed at 0.5, and the mean comprised an intercept and covariate $\mathbf{x}_1$ drawn from $N(0, 1)$. 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, 5-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.

(a) RMSPE   (b) CRPS
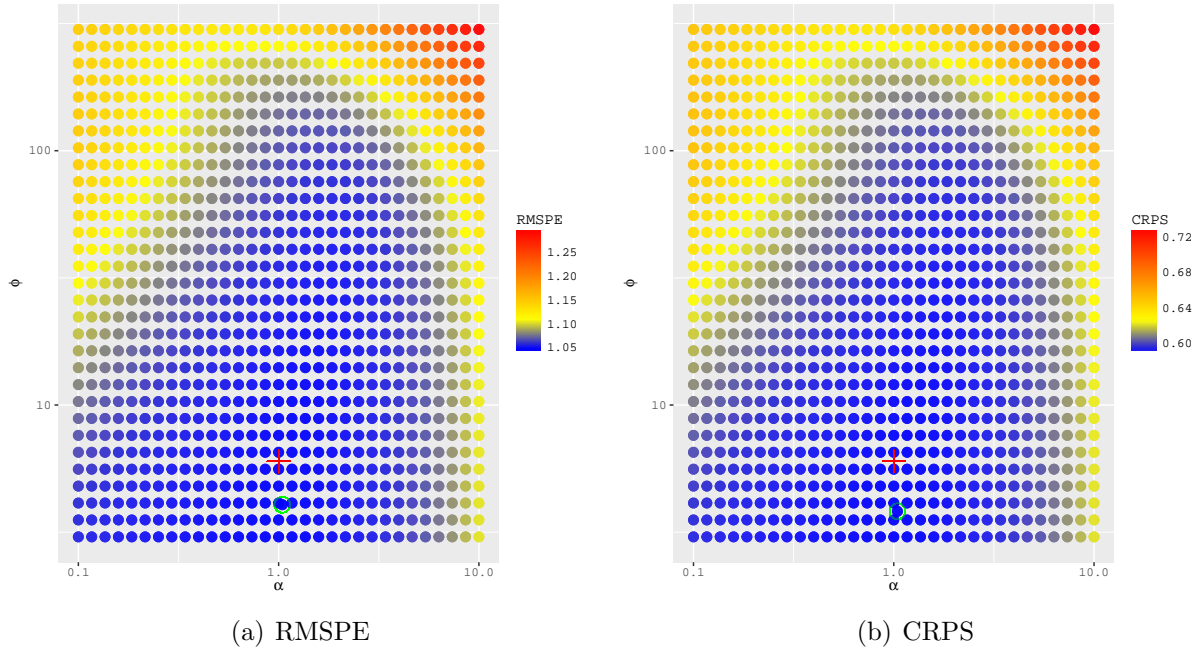
Figure 3: Conjugate model cross-validation results for selection of α and ϕ using the simulated dataset. Parameter combination with minimum scoring rule indicated with open circle symbol ○ and true combination used to generate the data indicated with a plus symbol +.

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 pre-

dictive 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).

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	–
95% PIW		3.87	3.85	3.93	3.84	–

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

For remote forested regions, combining airborne LiDAR data with a sparse network of forest inventory data provides a cost-effective means to deliver predictive maps of forest resources. In this study, LiDAR data were acquired across the TIU in Interior Alaska, approximately 140,000 km^2, using the G-LiHT airborne imager described in Section 1. 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
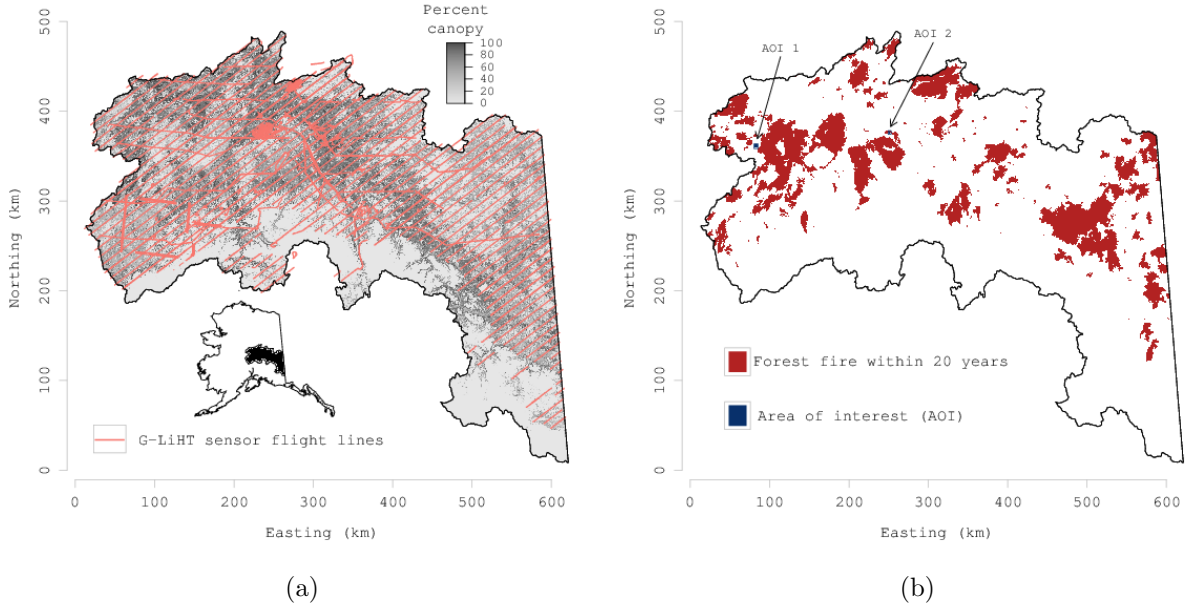
Figure 4: TIU, Alaska, study region. (a) G-LiHT flight lines where canopy height was measured at 5×10^6 locations and percent tree cover predictor variable. (b) Forest fire occurrence predictor variable and areas of interest for prediction illustration.

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 (i.e., the distance where the spatial correlation is 0.05) between 0.3 to 30 km. Observations at $n=5 \times 10^6$ locations, selected at random, were used to estimate model parameters.

25

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.

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	–
95% PIW	16.27	6.58	6.56	–
Run time (hours)	–	38.29	318.81	0.002

Figure 4(b) identifies two areas of interest (AOI) used to illustrate 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 AOIs. 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
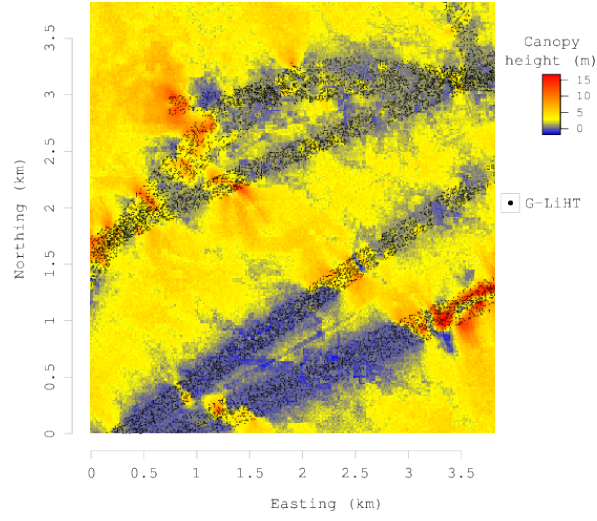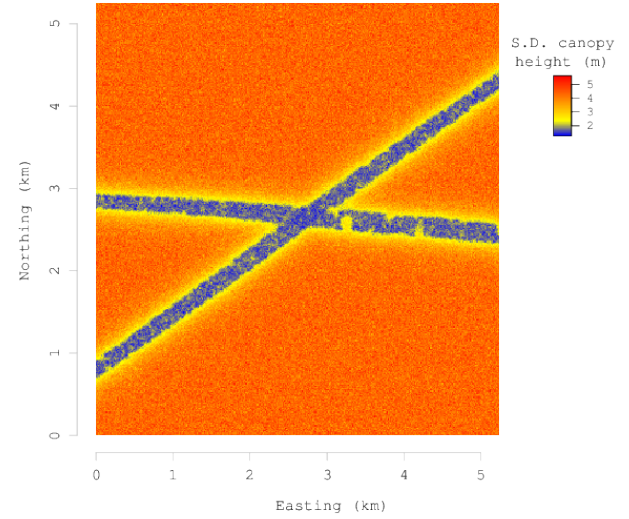
(a) AOI 1, posterior predictive mean
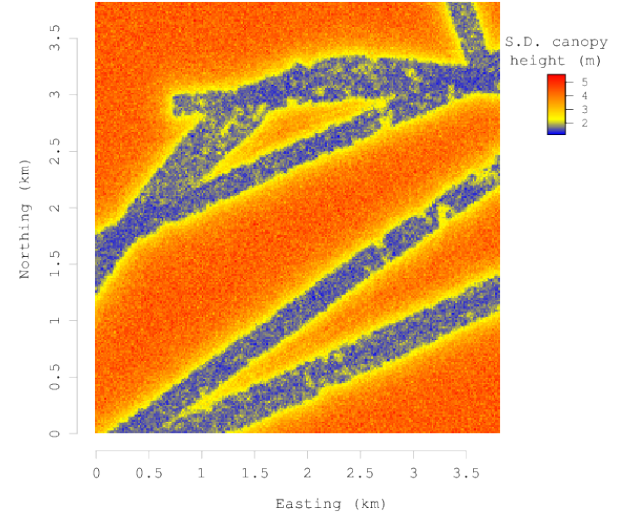
(b) AOI 1, posterior predictive S.D.
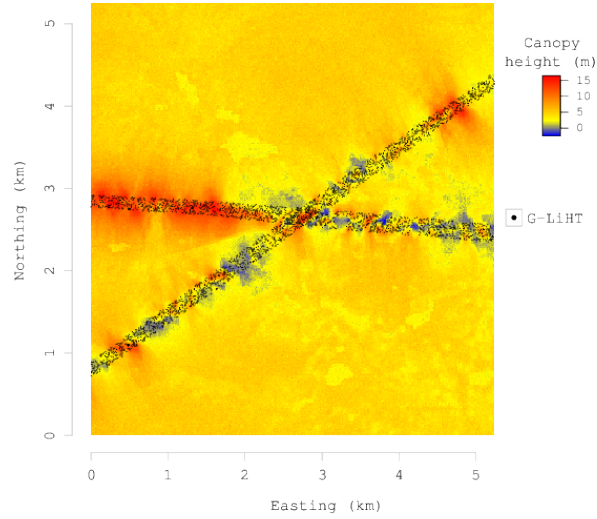
(c) AOI 2, posterior predictive mean

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

Figure 5: 95th LiDAR percentile height posterior predictive distribution summary at a 30 m pixel resolution for the two areas of interest identified in Figure 4(b).

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

27

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. As described in Section 1, next generation LiDAR instruments capable of global-scale mapping will sample forest characteristics using LiDAR instruments. 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 $\sim 17\%$ ($5 \times 10^6/28{,}751{,}400$) 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 and GEDI, when they choose between pules density and swath width.

4   Summary

Our aim has been to propose alternate formulations and derivatives of NNGP models developed by Datta et al. (2016) to substantially improve computational efficiency for fully process-based Bayesian 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 effort. Analysis of simulated data shows that compared with the Sequential specification of Datta et al. (2016), 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, the Collapsed model should generally be selected over the Response model. However, 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 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., 2016,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 exciting avenue for additional 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 (2016) suggest that the Kullback-Leibler divergence between NNGP models and full Gaussian process models can be sensitive to topological ordering and neighbor selection. 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. Nevertheless, we are currently conducting further investigations with the neighbor selection algorithms proposed by Guinness (2016) and intend to report on our findings in a subsequent report. 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 or Graphical Gaussian models and assessing their efficiency for highly complex multivariate spatial datasets.

Acknowledgments

References

Abdalati, W., Zwally, H., Bindschadler, R., Csatho, B., Farrell, S., Fricker, H., Harding, Dand 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, P. R., Davis, T. A., and Duff, I. S. (1996), "An Approximate Minimum Degree Ordering Algorithm," *SIAM Journal on Matrix Analysis and Applications*, 17, 886–905.

— (2004), "Algorithm 837: AMD, an Approximate Minimum Degree Ordering Algorithm," *ACM Transactions on Mathematical Software*, 30, 381–388.

Baccini, A., Friedl, M. A., Woodcock, C. E., and Warbington, R. (2004), "Forest biomass estimation over regional scales using multisource data," *Geophysical Research Letters*, 31, n/a–n/a, l10501.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014), *Hierarchical Modeling and Analysis for Spatial Data*, Boca Raton, FL: Chapman & Hall/CRC, 2nd ed.

Breiman, L. (2001), "Random Forests," *Machine Learning*, 45, 5–32.

Chen, Y., Davis, T. A., Hager, W. W., 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.

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, A. O., and Gelfand, A. E. (2016a), "Hierarchical Nearest-Neighbor Gaussian Process Models for Large Geostatistical Datasets," *Journal of the American Statistical Association*, 111, 800–812.

— (2016b), "On Nearest-Neighbor Gaussian Process Models For Massive Spatial Data," *Wiley Interdisciplinary Reviews: Computational Statistics*, 8, 162–171.

Datta, A., Banerjee, S., Finley, A. O., Hamm, N. A. S., and Schaap, M. (2016c), "Nonseparable 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.

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

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

Fayad, I., Baghdadi, N., Bailly, J.-S., Barbier, N., Gond, V., Hérault, B., El Hajj, M., Fabre, F., and Perrin, J. (2016), "Regional Scale Rain-Forest Height Mapping Using Regression-Kriging of Spaceborne and Airborne LiDAR Data: Application on French Guiana," *Remote Sensing*, 8, 240.

Finney, M. A. (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.

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.

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

Guinness, J. (2016), "Permutation Methods for Sharpening Gaussian Process Approximations," *https://arxiv.org/abs/1609.05372*.

Hager, W. W. (2002), "Minimizing the Profile of a Symmetric Matrix," *SIAM Journal on Scientific Computing*, 23, 1799–1816.

Hansen, M. C., Potapov, P. V., Moore, R., Hancher, M., Turubanova, S. A., Tyukavina, A., Thau, D., Stehman, S. V., Goetz, S. J., Loveland, T. R., Kommareddy, A., Egorov, A., Chini, L., Justice, C. O., and Townshend, J. R. G. (2013), "High-Resolution Global Maps of 21st-Century Forest Cover Change," *Science*, 342, 850–853.

Hurtt, G. C., Dubayah, R., Drake, J., Moorcroft, P. R., Pacala, S. W., Blair, J. B., and Fearon, M. G. (2004), "Beyond Potential Vegetation: Combining Lidar Data and a Height-Structured Model for Carbon Studies," *Ecological Applications*, 14, 873–883.

ICESat (2016), "Ice, Cloud, and Land Elevation Satellite," `http://icesat.gsfc.nasa.gov/icesat/glas.php`, accessed: 1-5-2015.

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. (2016), "A Multi-Resolution Approximation for Massive Spatial Datasets," *Journal of the American Statistical Association*, doi: 10.1080/01621459.2015.1123632.

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

Lauritzen, S. L. (1996), *Graphical Models*, Oxford, United Kingdom: Clarendon Press.

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

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

Mateu, J. and Müller, W. (2012), *Spatio-Temporal Design: Advances in Efficient Data Acquisition*, John Wiley & Sons, Ltd.

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

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

Sellers, P. J., Dickinson, R. E., Randall, D. A., Betts, A. K., Hall, F. G., Berry, J. A., Collatz, G. J., Denning, A. S., Mooney, H. A., Nobre, C. A., Sato, N., Field, C. B., and Henderson-Sellers, A. (1997), "Modeling the Exchanges of Energy, Water, and Carbon Between Continents and the Atmosphere," *Science*, 275, 502–509.

Simard, M., Pinto, N., Fisher, J. B., and Baccini, A. (2011), "Mapping forest canopy height globally with spaceborne lidar," *Journal of Geophysical Research: Biogeosciences*, 116, n/a–n/a, g04021.

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*, 66, 275–296.

Stratton, R. D. (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, J. R., Stein, M. L., and Lysen, S. (2014), "Bayesian and Maximum Likelihood Estimation for Gaussian Processes on an Incomplete Lattice," *http://arxiv.org/abs/1402.4281*.

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.

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

Xia, G., Miranda, M. L., and Gelfand, A. E. (2006), "Approximately Optimal Spatial Design Approaches for Environmental Health Data," *Environmetrics*, 17, 363–385.

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.

Web supplement for *Applying Nearest Neighbor Gaussian Processes to Massive Spatial Data Sets: Forest Canopy Height Prediction Across Tanana Valley Alaska*

S1 Fast nearest neighbor search

Construction of NNGP models require a neighbor set for each observed location. Identifying these sets is trivial when n is small. For a given location $\mathbf{s}_i$, a brute force approach to finding the neighbor set calculates the 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. Figure S1 shows the time required to perform this *brute force* approach for a range of n. Here, we can see that when n is larger than $\sim 5 \times 10^4$ the brute force approach becomes prohibitively slow. Distributing this brute force search over multiple CPUs, e.g., 12 CPUs also shown in Figure S1, does improve performance; however, n larger than $\sim 5 \times 10^5$ is again too slow. Developing data structures and associated Algorithms for efficient nearest neighbor searches is a major focus in computer science and engineering, see, e.g., Bühlmann et al. (2016) Ch. 7. Given the size of datasets considered in subsequent analyses, i.e., $< 1 \times 10^7$, we chose 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 as shown in Figure S1 (labeled *fast*). We note that future work could look into modifying more sophisticated structures and associated search Algorithms such as binary search trees (Cormen et al., 2009) to deliver the nearest neighbor set more efficiently given the NNGP ordering constraint.
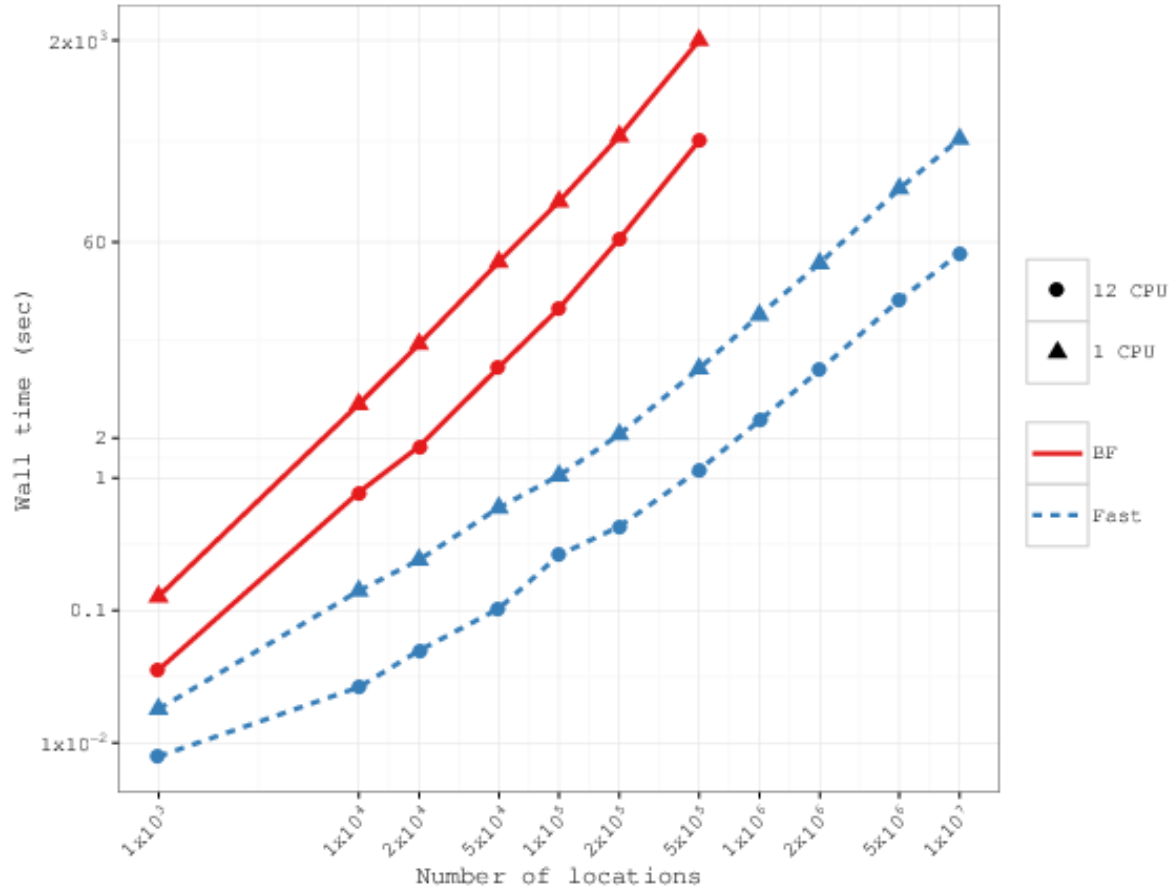
Figure S1: Wall time required for neighbor set search using a brute force (BF) and fast search Algorithm by n and number of CPUs.
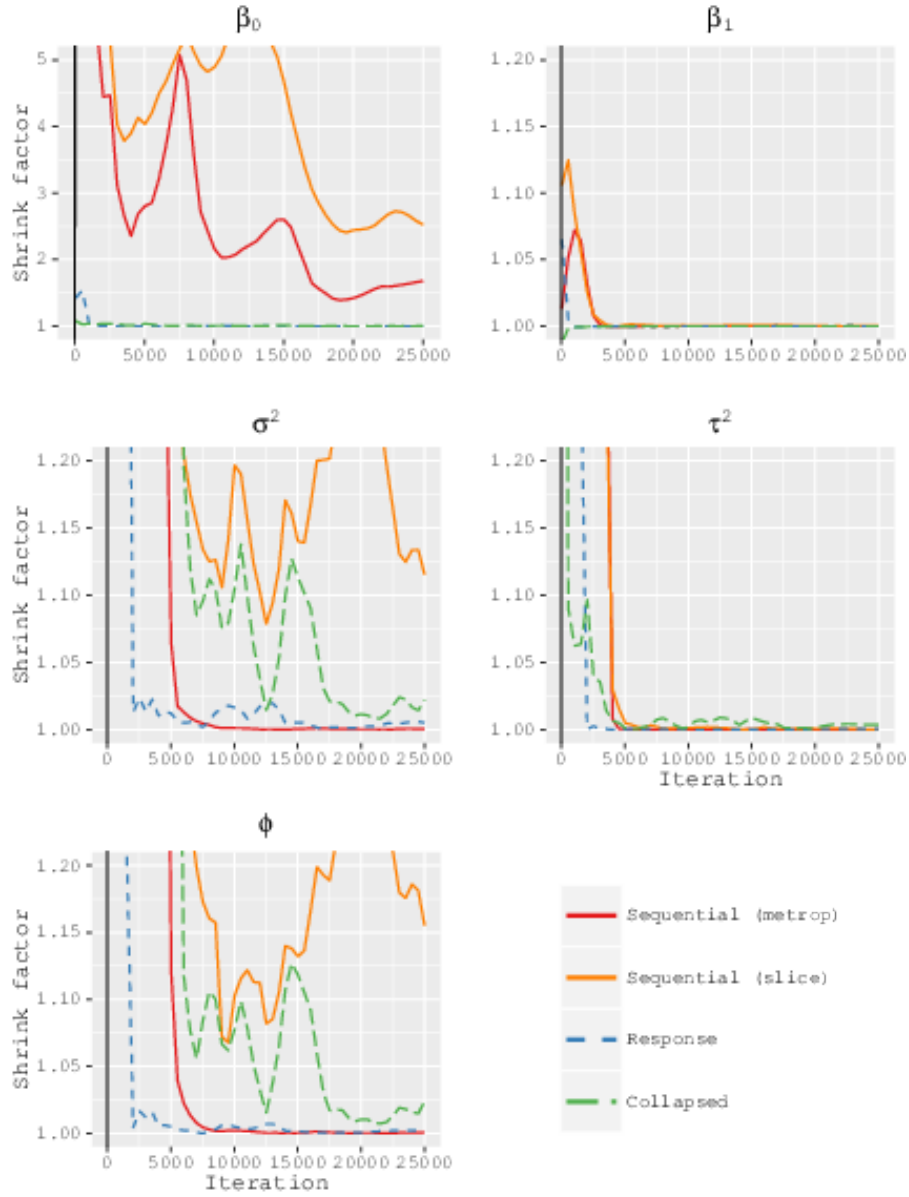
S2   Experiment #2

Figure S2: Simulated dataset, Gelman-Rubin convergence diagnostics plots for candidate model parameters. Results consider both a Metropolis Hastings (metrop) and Slice sampler (slice) update of the Response models' covariance parameters.
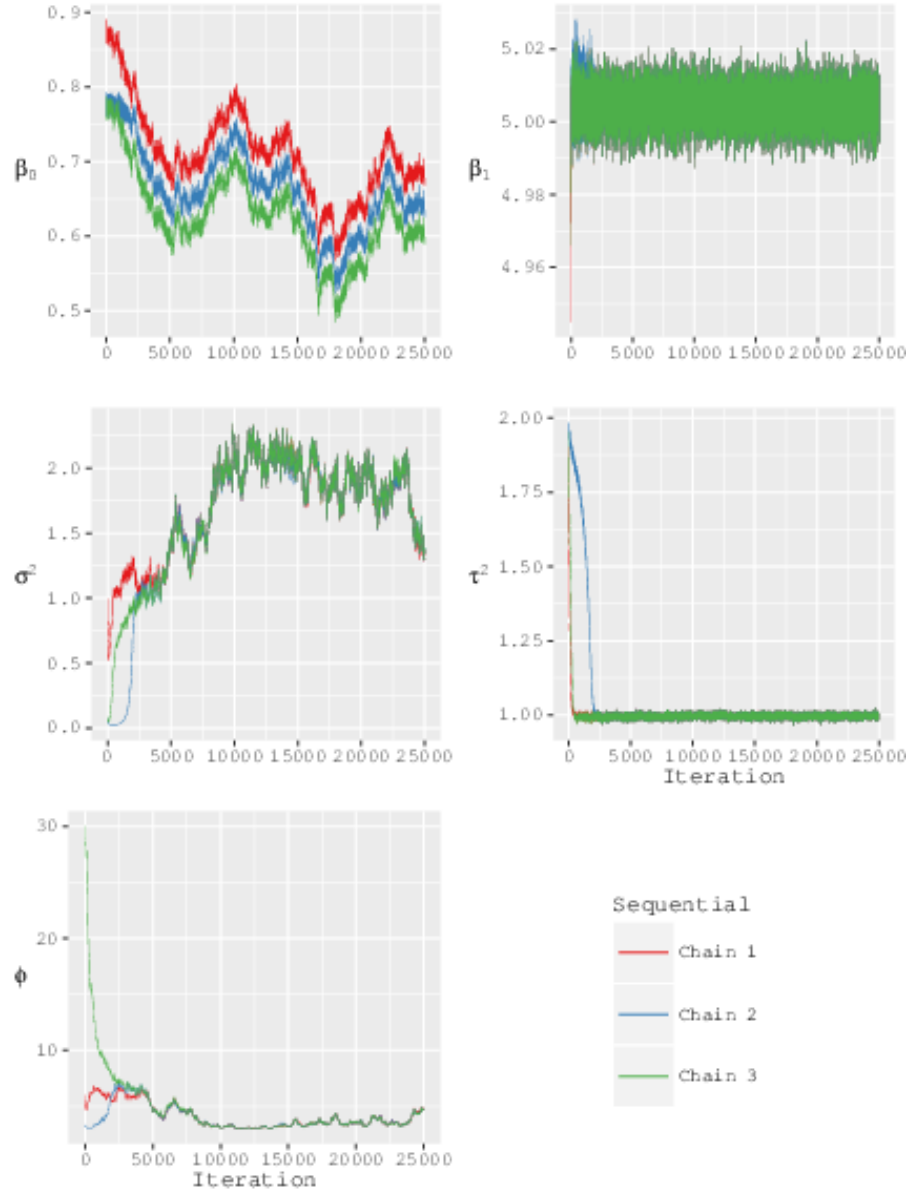
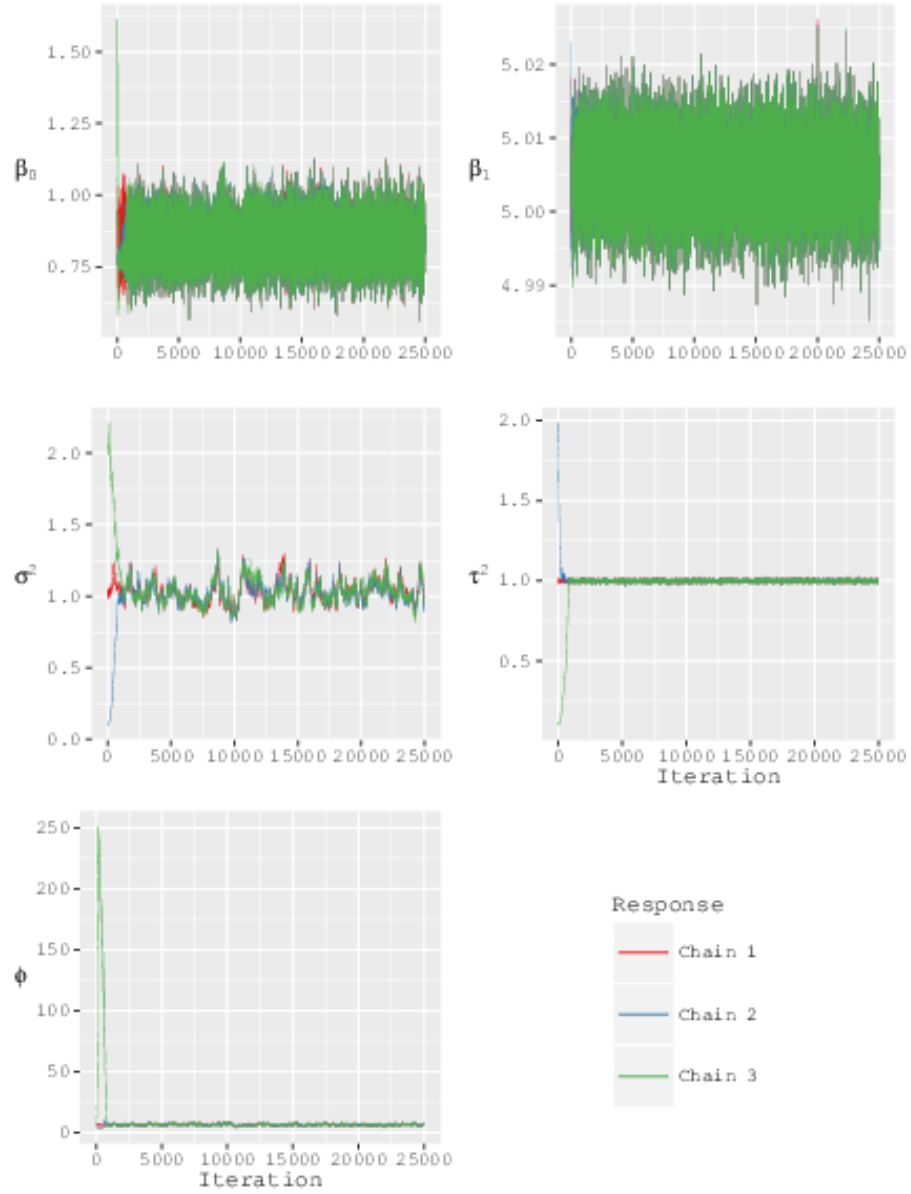Figure S3: Simulated dataset, Sequential model MCMC chain trace plots.

Figure S4: Simulated dataset, Response model MCMC chain trace plots.
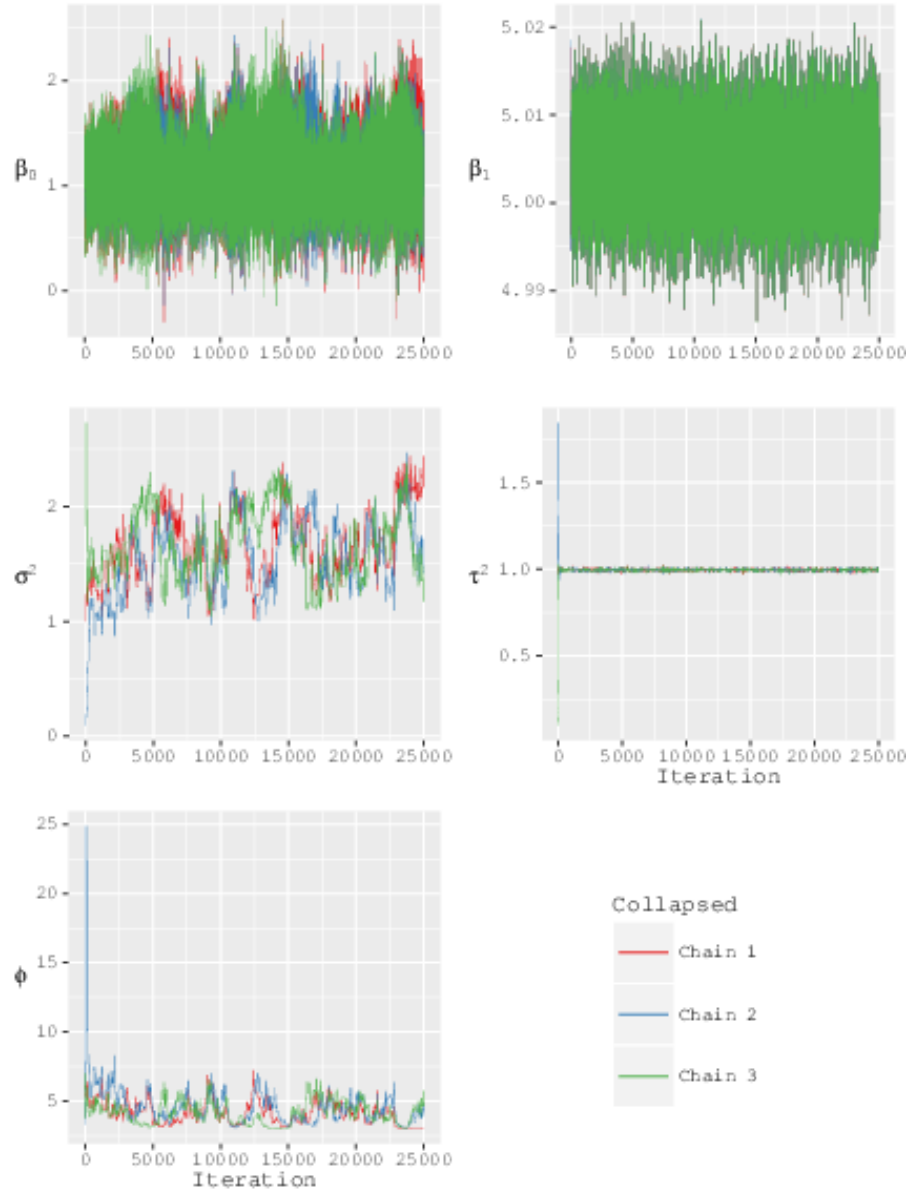
Figure S5: Simulated dataset, Collapsed model MCMC chain trace plots.

S3   Experiment # 3

Simulated data were generated from the model specified in Experiment #2 detailed in Section 3.3 with the exception that $n = 1500$. Observations at $n = 1000$ of these locations, selected at random, were used to estimate model parameters. Observations at the remaining 500 holdout locations were used to assess model predictive performance. Specifically, given the holdout observations and model posterior predictive distribution samples, predictive performance was summarized using: 1) mean continuous rank probability score (CRPS), which is a strictly proper scoring rule that quantifies the fit of the entire predictive distribution (i.e., for a normal distribution, the mean and the variance) to the data (Gneiting and Raftery, 2007); 2) root mean-square prediction error (RMSPE) between observed values and means of the predictive distributions; 3) percent of observations covered by their corresponding predictive distribution 95% credible interval (PCI), and; mean width of the predictive distributions' 95% credible interval (PIW).

The experiment sample size was kept purposely small so we could compare NNGP model performance to that of a full GP model. Full GP model parameter estimates and predictions were obtained using the spBayes R package `spLM` function (Finley et al., 2015). For all models, the prior distribution on regression coefficients β_0 and β_1 were assumed to be *flat* and variance parameters σ^2 and τ^2 followed an inverse-gamma $IG(2, 1)$. For the full GP, Sequential, Response, and Collapsed models the spatial decay parameter ϕ followed a uniform $U(3, 300)$. This prior support for ϕ assumes the effective range is between 0.01 and 1 distance units (where the effective range is defined as the distance at which the correlation equals 0.05). As detailed in Section 1.4, the Conjugate model ϕ and α were selected to minimize RMSPE using a 5-fold cross-validation.

The aim of this experiment was to assess samplers' convergence characteristics and determine to what extent NNGP model inference compares to the full GP model. The Gelman-Rubin (Gelman and Rubin, 1992) potential scale reduction factor, Figure S6, and visual inspection of MCMC chain trace plots Figures S8-S10 showed adequate convergence and

mixing within a thousand MCMC iterations for all models. Parameter estimates and predictive performance metrics in Table S1 suggest that for these data, NNGP models do indeed deliver inference comparable to the full GP model. Results from more extensive simulation experiments conducted by Datta et al. (2016) and ? comparing the Sequential and full GP models corroborate our findings. Quantile-quantile (Q-Q) plots in Figure S7 provide a more detailed comparison between the posterior distributions generated using the NNGP and full GP models. Here, the Sequential model's β_0 shows the most striking departure from the full GP estimates—with tails substantially shorter than those of the full GP. There were some other differences in the tails of the NNGP posterior distributions compared to the full GP model estimates; however, these were relatively minor. Increasing the nearest neighbor set size from 15 to 25 did not have a substantial impact on the NNGP model posterior distributions or predictive inference (Table S1).

Table S1: Simulated dataset, model parameter estimates $50\%\,(2.5\%, 97.5\%)$ and predictive validation.

| Parameter	True	Full GP	Sequential (m=15)	Response (m=15)	Collapsed (m=15)	Conjugate (m=15)
β_0	1	1.28 (0.65, 1.99)	1.25 (0.74, 1.92)	1.25 (0.56, 1.91)	1.24 (0.50, 1.93)	1.28
β_1	5	4.99 (4.93, 5.05)	4.99 (4.93, 5.06)	4.99 (4.93, 5.06)	4.99 (4.93, 5.06)	4.99
σ^2	1	1.11 (0.77, 1.85)	1.10 (0.76, 1.79)	1.20 (0.80, 1.99)	1.18 (0.78, 1.95)	1.17
τ^2	1	0.96 (0.86, 1.10)	0.97 (0.85, 1.09)	0.97 (0.85, 1.10)	0.97 (0.85, 1.10)	0.9
ϕ	6	6.06 (3.34, 9.71)	6.08 (3.40, 9.61)	5.45 (3.20, 9.31)	5.53 (3.18, 9.50)	7.89
CRPS		0.65	0.66	0.66	0.66	0.65
RMSPE		1.15	1.15	1.16	1.15	1.15
95% PIC		92.8	93.8	92.4	92	–
95% PIW		4.17	4.32	4.18	4.09	–
		Sequential (m=25)	Response (m=25)	Collapsed (m=25)	Conjugate (m=25)	
β_0	1	1.27 (0.78, 1.94)	1.27 (0.58, 1.95)	1.27 (0.58, 1.94)	1.28	
β_1	5	4.99 (4.93, 5.06)	4.99 (4.93, 5.06)	4.99 (4.93, 5.06)	4.99	
σ^2	1	1.05 (0.74, 1.71)	1.08 (0.74, 1.79)	1.12 (0.75, 1.83)	1.1	
τ^2	1	0.97 (0.85, 1.09)	0.98 (0.86, 1.12)	0.97 (0.85, 1.10)	0.92	
ϕ	6	6.42 (3.74, 9.86)	5.87 (3.26, 9.51)	5.92 (3.30, 9.73)	7.89	
CRPS		0.66	0.66	0.66	0.65	
RMSPE		1.15	1.16	1.15	1.14	
95% PIC		94	92.4	92.4	–	
95% PIW		4.32	4.18	4.10	–	
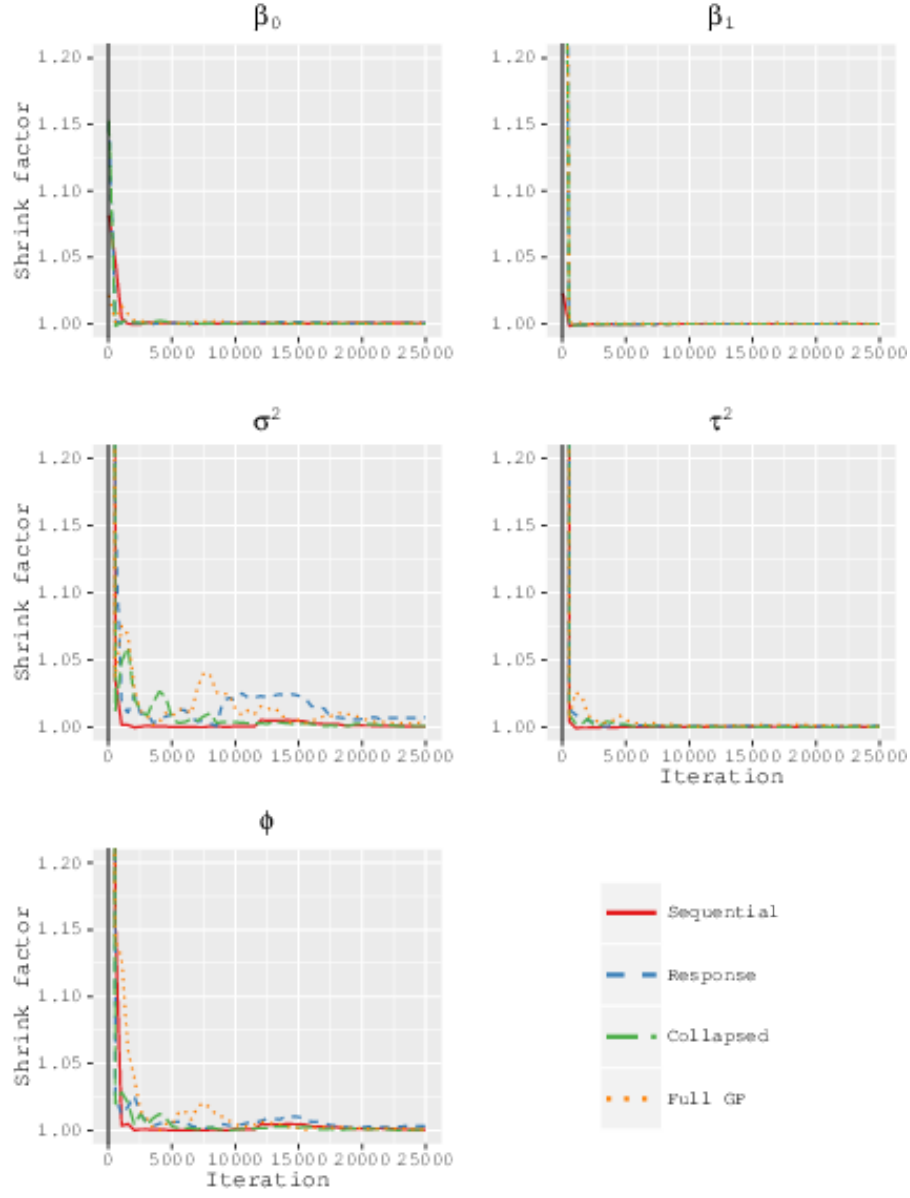
Figure S6: Simulated dataset, Gelman-Rubin convergence diagnostic plots for NNGP and full GP model parameters.
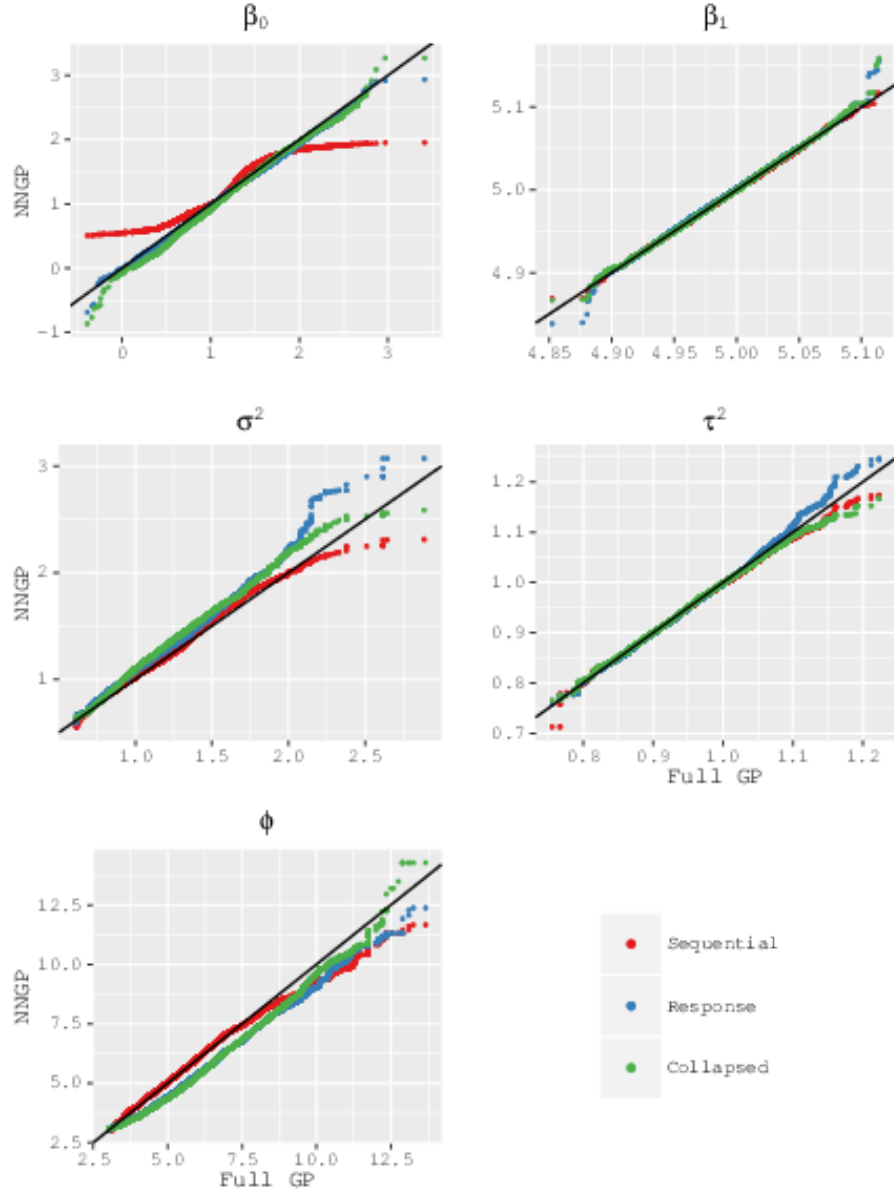
Figure S7: Simulated dataset, quantile-quantile plots of NNGP versus full GP model parameter posterior distributions.
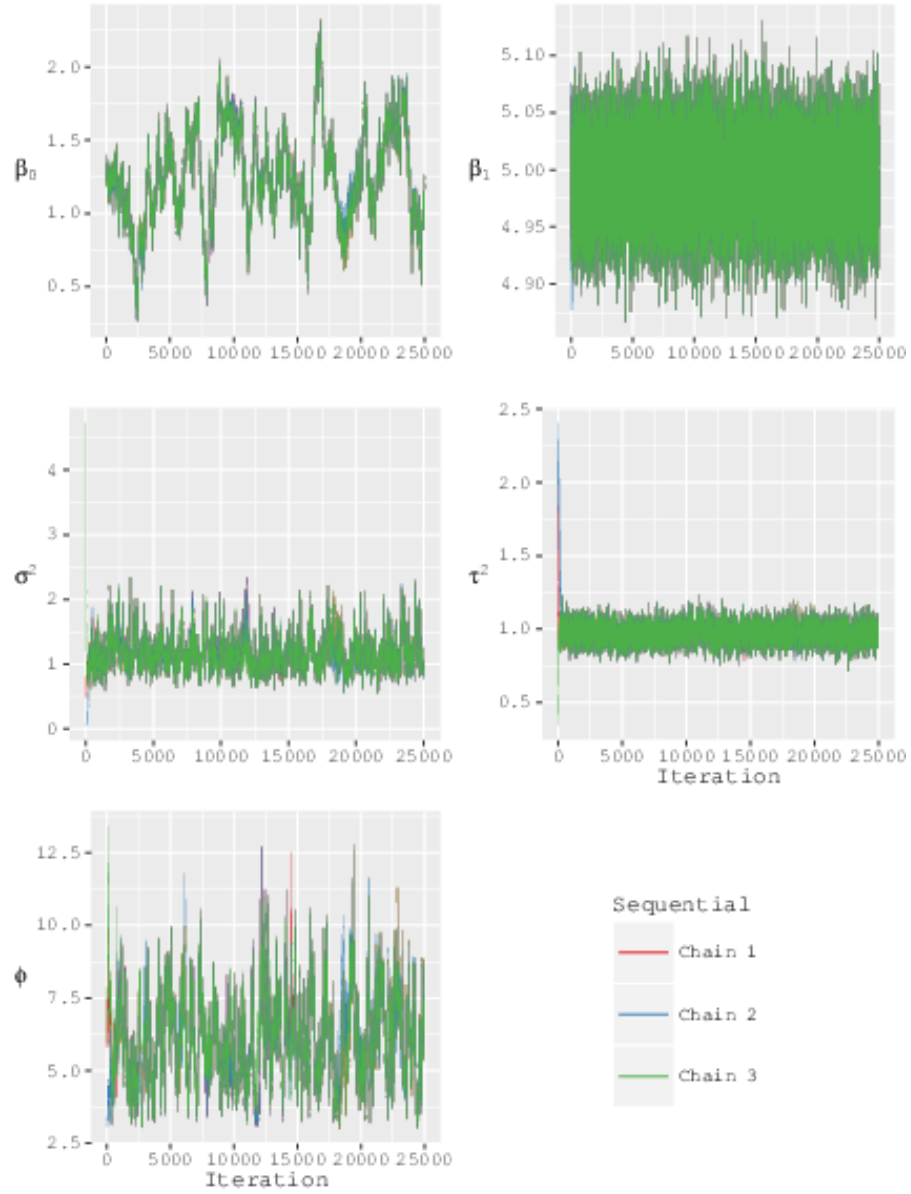
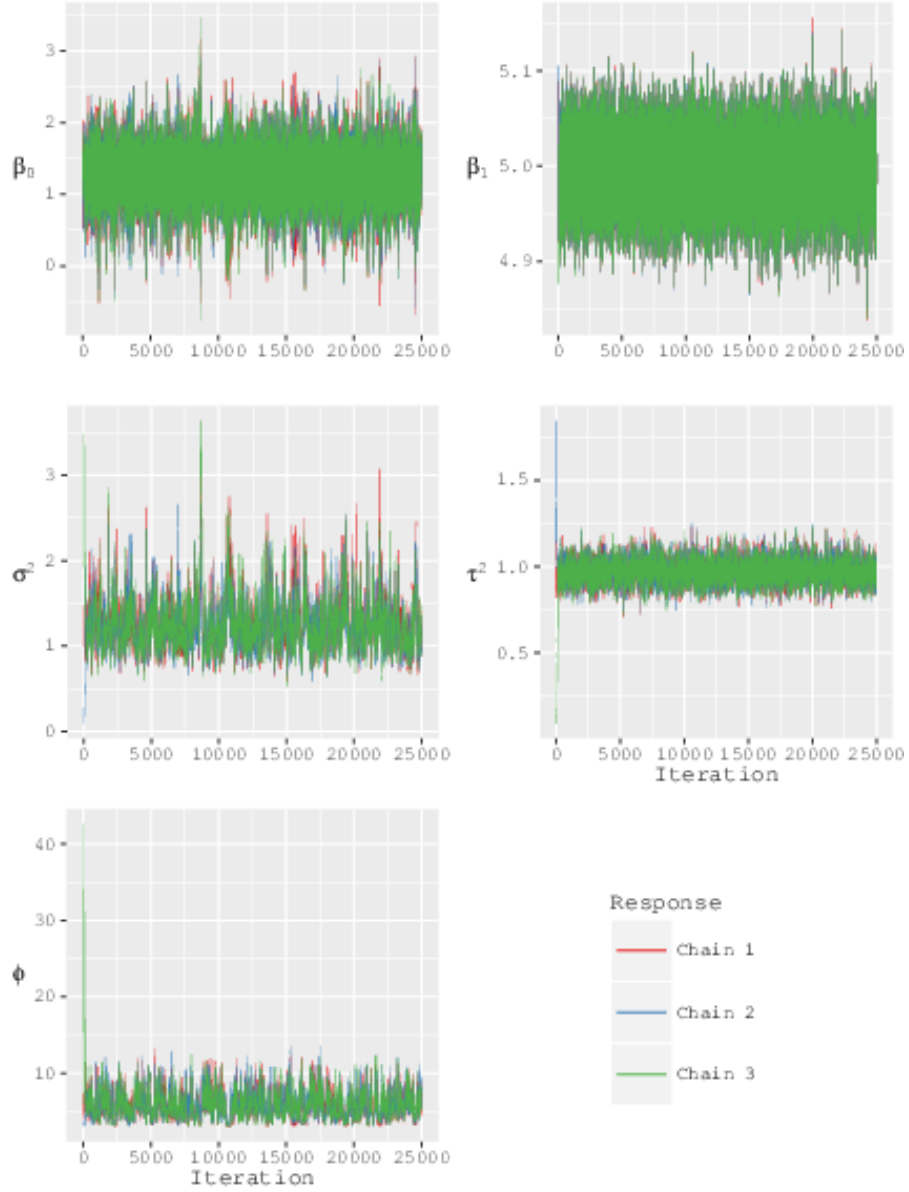Figure S8: Simulated dataset, Sequential $m=15$ model MCMC chain trace plots.

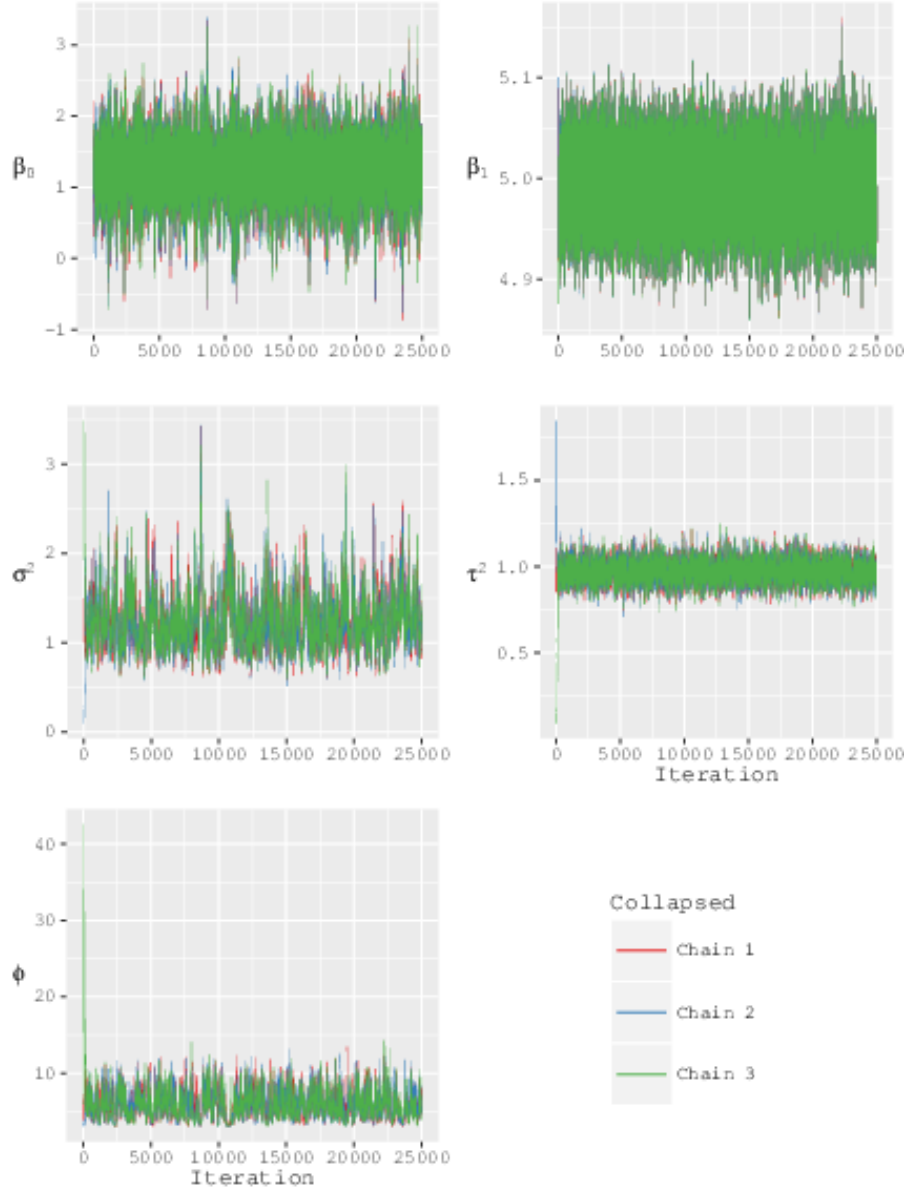Figure S9: Simulated dataset, Response $m=15$ model MCMC chain trace plots.

Figure S10: Simulated dataset, Collapsed $m=15$ model MCMC chain trace plots.

References

Bühlmann, P., Drineas, P., Kane, M., and van Der Laan, M. (2016). *Handbook of Big Data.* Handbooks of Modern Statistical Methods. Chapman & Hall/CRC.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms, Third Edition.* The MIT Press, 3rd edition.

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(514):800–812.

Finley, A., Banerjee, S., and Gelfand, A. (2015). spbayes for large univariate and multivariate point-referenced spatio-temporal data models. *Journal of Statistical Software*, 63(1):1–28.

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

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378.

Ra, S. W. and Kim, J. K. (1993). Fast mean-distance-ordered partial codebook search algorithm for image vector quantization. *IEEE Transactions on Circuits and Systems II*, 40(9):576–579.