Distributed Implementation of Nearest-Neighbor Gaussian Processes

Isabelle Grenier*and Bruno Sansó †
Department of Statistics, University of California, Santa Cruz

September 12, 2019

Abstract

While many statistical approaches have tackled the issue of large spatial datasets, the issue arising from costly data movement and data storage have long been set aside. Having an easy access to the data has been taken for granted and is now becoming an important bottleneck in the performance of statistical inference. As the availability of high resolution spatial data continues to grow, the need to develop an efficient modeling technique is becoming a priority. In this paper, we develop a distributed method for the Nearest-Neighbor Gaussian Process (NNGP) models as a solution to large datasets. The framework that we propose retain the exact implementation of the NNGP while allowing for a parallel computation of the posterior inference. The method allows for any choice of grouping of the data whether it is at random or by region. As a result of this new method, the NNGP model can be applied to a dataset with n observations split into J servers with computations of order n/J.

Keywords: Bayesian methods, Computationally-intensive methods, Spatial Analysis, Statistical Computing

*igrenier@ucsc.edu
†bruno@soe.ucsc.edu. The authors were partially funded by the National Science Foundation Grant DMS-1513076

1 Introduction

The increased availability of georeferenced data has produced a need to handle, analyze and make inferences and predictions for very large collections of data, posing a large computational burden on model-based inference of spatial fields. Geostatistical methods that deal with point-referenced data consider random fields that are indexed in space, usually 2D or 3D. Intuitively, proximity between observations should provide information for inference about unobserved values of the field. This is often formalized using model-based approaches for which there is a solid body of literature and software, see, for example, the books by Cressie (1993); Gelfand et al. (2010); Cressie and Wikle (2011); Banerjee et al. (2014). Modern geostatistical approaches provide flexible probabilistic models, coupled with powerful learning methods, that are used to investigate challenging inferential questions related to geographically-referenced data. Traditionally, model-based spatial models have relied on the Gaussian process (GP). GPs capture the dependence due to proximity trough a covariance function. For a likelihood-based approach to GPs, the bottleneck lies in the computation of the determinant and the inverse of the covariance matrix induced by the locations of the available observations. Matrix inversion has order n^3 operations, which can be costly for datasets with large number of observations n. To illustrate the problem with some numbers, an application for observations at, say, 10,000 locations will produce a covariance matrix with 50,005,000 possibly different values. Such a large data structure would need to be stored, decomposed and operated with, possibly within an iterative procedure. Modern applications can have datasets that are orders of magnitude larger than that. To tackle this problem most current methods take one of two approaches: exploit sparsity in the structure of the covariance matrix or reduce the dimensionality of the problem by seeking representations of GPs on lower dimensional subspaces. In both cases the goal is to speed up calculations, as well as reduce the size of the objects that need to be handled and stored in memory when performing computations. An example of the former is covariance tapering that consists of truncating the covariance function to zero for distant observations (Furrer et al., 2006). By using an appropriate tapered correlation kernel, the

covariance matrix becomes very sparse while remaining positive definite. An example of the latter is the predictive Gaussian processes, as introduced by Banerjee et al. (2008), which represents the GP using basis functions generated by the covariance function. This produces a reduction of the dimension of the matrix that needs to be inverted to perform inference to a fixed, small dimension, that depends on a pre-specified set of knots. This is a similar approach to the one in Higdon (1998); Cressie and Johannesson (2008). For further information about the state of the art model-based geostatistics methods suitable for large data sets see Banerjee (2017) and Heaton et al. (2018). In this paper we focus on nearest-neighbor GP ((NNGP), Datta et al., 2016), a model that is particularly intriguing, as it blends features of both the dimension reduction and the sparsity approaches. NNGP uses the conditional distribution structure of a joint likelihood to build a directed network of neighbors . By making observations conditionally independent of non-neighboring locations, the new precision matrix has a very sparse structure. In that case, the number of operations needed to invert the precision matrix is limited by the number of neighboring locations allowed.

While data storage has been improved to accomodate the flood of information needed to be stored, model-based approaches rely on data access and ultimately on data movement. In most cases, models take for granted that the data can be accessed and manipulated all at once. In recent years, several situations have emerged where the data have been stored in multiple locations (distributed data). Similarly, data may be stored in one location but be too large to use at once. Under these scenarios, our primary objective is to adapt our statistical methods to keep up with competitive algorithms. Distributed computing and parallel implementation are key to the next significant gain in efficiency of statistical inference. In this paper, we introduce a divide-and-conquer approach for NNGPs. The gain in efficiency from the distributed approach is of particular interest as it does not sacrifice accuracy in the process. In section 2, we review the distributed approaches suggested by Katzfuss and Hammerling (2014) and Guhaniyogi et al. (2017). In section 3, we introduce NNGPs and detail our divide-and-conquer strategy. Finally, in the last section, we illustrate the method by applying it to a simulated dataset and the soil moisture active-passive

(SMAP) satellite data for three days in August 2017.

2 Distributed Computing for Spatial Models

Previous work in distributed approaches for statistical models include the parallelization of low-rank models by Katzfuss and Hammerling (2014), and the aggregation of Bayesian posterior inference by Guhaniyogi et al. (2017). The goal of this section is to discuss the current state of distributed computing for spatial inference and motivate the need for our new approach.

2.1 Parallel Inference for Low-Rank Models

While low-rank models already achieve a significant improvement in computational efficiency compared to the implementation of the full GP, Katzfuss and Hammerling (2014) took a step further by using parallelization to achieve an even greater speed gain. Their divide-and-conquer approach was developed with two scenarios in mind. The first situation assumes that the data reside on J servers. In that case, moving the data to a common server is too slow but moving the results on the other hand, is fast. The second situation applies to any large spatial dataset which can benefit from being separated into J blocks. In either case, the studied model is such that

$$y(\boldsymbol{s}_{j,i}) = w(\boldsymbol{s}_{j,i}) + \epsilon(\boldsymbol{s}_{j,i}) \tag{1}$$

where $\boldsymbol{s}_{j,i}$ is the location of observation i on server j for $i = 1, ..., n_j$, $j = 1, ..., J$, and $\epsilon(\boldsymbol{s}_{j,i}) \sim N(0, v_\epsilon(\boldsymbol{s}_{j,i}))$ is independent of y for a known function v_ϵ. In spatial low-rank models, the approximation of the true underlying process is based on a set of m basis functions $\boldsymbol{B}$:

$$w(\boldsymbol{s}_{j,i}) = \boldsymbol{B}(\boldsymbol{s}_{j,i})^{'}\boldsymbol{\eta} + \delta(s_{j,i}),$$

where $\delta \sim N(0, v_\delta(\boldsymbol{s}_{j,i}))$ is spatially independent and independent of $\boldsymbol{\eta}$. Assuming the covariance parameters are fixed, and using a normal prior, $\boldsymbol{\eta} \sim N_m(\boldsymbol{\nu}_0, \boldsymbol{K}_0)$, the posterior

distribution of $\boldsymbol{\eta}$ is $N_m(\boldsymbol{\nu}_y, \boldsymbol{K}_y)$, where

$$\begin{aligned} \boldsymbol{K}_y^{-1} &= \boldsymbol{K}_0^{-1} + \boldsymbol{R}, \;\; \boldsymbol{R} = \boldsymbol{B}_{1:J}^{'} \boldsymbol{V}_{1:J}^{-1} \boldsymbol{B}_{1:J} \\ \boldsymbol{\nu}_y &= \boldsymbol{K}_y(\boldsymbol{K}_0^{-1}\boldsymbol{\nu}_0 + \boldsymbol{\gamma}), \;\; \boldsymbol{\gamma} = \boldsymbol{K}_{1:J}^{'} \boldsymbol{K}_{1:J}^{-1} y_{1:J}, \end{aligned}$$

where $\boldsymbol{B}_{1:J} = (\boldsymbol{B}_1, ..., \boldsymbol{B}_J)$ is a vector of matrices where each $\boldsymbol{B}_j = (\boldsymbol{B}(\boldsymbol{s}_{j,1}), ..., \boldsymbol{B}(\boldsymbol{s}_{j,n_j}))$ and $\boldsymbol{V}_{1:J} = blockdiag(\boldsymbol{V}_1, ..., \boldsymbol{V}_J)$ where each $\boldsymbol{V}_j = diag(v_\delta(\boldsymbol{s}_{j,1}) + v_\epsilon(\boldsymbol{s}_{j,1}), ..., v_\delta(\boldsymbol{s}_{j,n_j}) + v_\epsilon(\boldsymbol{s}_{j,n_j}))$. Because of the block structure of $\boldsymbol{V}_{1:J}$, the above calculations become a sum of quantities that can be computed independently on each server:

$$\boldsymbol{R} = \sum_{j=1}^{J} \boldsymbol{B}_j^{'} \boldsymbol{V}_j^{-1} \boldsymbol{B}_j \;, \;\; \boldsymbol{\gamma} = \sum_{j=1}^{J} \boldsymbol{B}_j^{'} \boldsymbol{V}_j^{-1} y_j.$$

The main algorithm is therefore reduced to computing the posterior parameters on each server j, moving the results to a central node and adding them to obtain the posterior parameters for the joint model. This is a special case of the algorithm developed by Qian (2018) discussed in section 3.2.

2.2 Distributed Kriging

Distributed Kriging (DISK) is another distributed approach for Bayesian modeling explored in Guhaniyogi et al. (2017). Let $w_{jb}(\boldsymbol{s}_i)$ be the collection of MCMC samples for server $j = 1, ..., J$ for location $\boldsymbol{s}_i$. The DISK posterior estimates are obtained by approximating the Wasserstein barycenters of the samples, that is by averaging over the empirical quantiles of each samples. The strength of the framework is that it is agnostic to the choice of model, however an important drawback is the assumption that the subsets are created at random and contain locations for each region of the spatial domain. In many applications, it is often the case that data is stored by region, violating the assumption. In those situations, an initial randomization step would be required to proceed with the technique which can be costly and inefficient. A more desirable approach will tackle the dataset as presented without incurring additional data movement costs.

3 Nearest-Neighbor Gaussian Process

Let $w(\boldsymbol{s}) \sim GP(0, C_\theta)$ denote a zero-centered GP where $\boldsymbol{s}$ is any location in a space $\mathcal{D}$. The process relies on a valid covariance function C_θ, $\theta = (\sigma^2, \phi)$, which only depends on the distance between pairs of observations. Let $\mathcal{S} = \boldsymbol{s}_1, ..., \boldsymbol{s}_k$ be a set of reference locations, possibly involving a grid where the indexes $1, ..., k$ are tied to a specified ordering of the locations. Since this is a fixed subset of $\mathcal{D}$, $\boldsymbol{w}_\mathcal{S} \sim N_k(0, C_\theta(\mathcal{S}))$ where $C_\theta(\mathcal{S})$ is the covariance matrix for $\mathcal{S}$ associated with the covariance function C_θ. The joint distribution of $\mathcal{S}$ can be expressed using a chain of conditional distribution:

$$p(w_\mathcal{S}) = p(w(\boldsymbol{s}_1)) \prod_{i=2}^{k} p(w(\boldsymbol{s}_i)|w(\boldsymbol{s}_1), ..., w(\boldsymbol{s}_{i-1})).$$

The Vecchia approximation as introduced in Vecchia (1988) suggests that for a large i, the conditional distribution above includes superfluous information. It is therefore appropriate to restrict the conditional distribution to an approximation of order m based on the Euclidean distance between the locations. This likelihood approximation method implies that the locations that are closest to $\boldsymbol{s}_i$ influence the value of $w(\boldsymbol{s}_i)$ the most. Applying this to the full joint distribution, we obtain

$$\tilde{p}(\boldsymbol{w}_\mathcal{S}) = p(w(\boldsymbol{s}_1)) \prod_{i=2}^{k} p(w(\boldsymbol{s}_i)|w_{N(\boldsymbol{s}_i)}),$$

where $N(s_i) \subset \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_{i-1}\}$ which includes the m closest locations to $\boldsymbol{s}_i$. This sequential structure produced by the ordering of the reference locations creates a directed acyclic graph which guarantees a proper joint density. Using the conditional Normal distribution density, we obtain the following approximated joint distribution

$$\tilde{p}(\boldsymbol{w}_\mathcal{S}) = \prod_{i=1}^{k} N(w(\boldsymbol{s}_i)|\boldsymbol{B}_{\boldsymbol{s}_i}\boldsymbol{w}_{N(\boldsymbol{s}_i)}, \boldsymbol{F}_{\boldsymbol{s}_i}),$$

where

$$\begin{aligned} \boldsymbol{B}_{\boldsymbol{s}_i} &= C_\theta(\boldsymbol{s}_i, N(\boldsymbol{s}_i))C_\theta^{-1}(N(\boldsymbol{s}_i)) \\ \boldsymbol{F}_{\boldsymbol{s}_i} &= C_\theta(\boldsymbol{s}_i) - C_\theta(\boldsymbol{s}_i, N(\boldsymbol{s}_i))C_\theta^{-1}(N(\boldsymbol{s}_i))C_\theta(N(\boldsymbol{s}_i), \boldsymbol{s}_i), \end{aligned}$$

where $C_\theta(\boldsymbol{s}_i, N(\boldsymbol{s}_i))$ is a vector where each entry is the covariance between $\boldsymbol{s}_i$ and its neighbors $N(\boldsymbol{s}_i)$, $C_\theta(N(\boldsymbol{s}_i))$ is a symmetric matrix with elements corresponding to the covariance of each pair of neighbors, and $C_\theta(\boldsymbol{s}_i)$ is the variance of $\boldsymbol{s}_i$. In all cases, the covariances are fully specified by the covariance function C_θ. The resulting distribution for $\tilde{p}(\boldsymbol{w}_\mathcal{S})$ is a multivariate normal distribution with covariance matrix denoted $\tilde{C}_\theta(\mathcal{S})$.

Let $\boldsymbol{u}$ be any location in $\mathcal{D}$, and $N(\boldsymbol{u})$ be the set of m neighbors of $\boldsymbol{u}$ in $\mathcal{S}$. As detailed in Datta et al. (2016), given a parent spatial process and a fixed reference set $\mathcal{S}$, we can construct a new process over $\mathcal{D}$. In this case, the original process is $GP(0, C_\theta)$, therefore for a fixed set of observations $\mathcal{U} = \boldsymbol{u}_1, ..., \boldsymbol{u}_n$, the nearest neighbors density of $\boldsymbol{w}_\mathcal{U}$ conditional on $\boldsymbol{w}_\mathcal{S}$ is

$$\begin{aligned} \tilde{p}(\boldsymbol{w}_\mathcal{U}|\boldsymbol{w}_\mathcal{S}) &= \prod_{i=1}^{n} p(w(\boldsymbol{u}_i)|\boldsymbol{w}_{N(\boldsymbol{u}_i)}) \quad (2) \\ &= \prod_{i=1}^{n} N(w(\boldsymbol{u}_i)|\boldsymbol{B}_{\boldsymbol{u}_i}\boldsymbol{w}_{N(\boldsymbol{u}_i)}, \boldsymbol{F}_{\boldsymbol{u}_i}), \quad (3) \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{B}_{\boldsymbol{u}_i} &= C_\theta(\boldsymbol{u}_i, N(\boldsymbol{u}_i))C_\theta^{-1}(N(\boldsymbol{u}_i)) \\ \boldsymbol{F}_{\boldsymbol{u}_i} &= C_\theta(\boldsymbol{u}_i) - C_\theta(\boldsymbol{u}_i, N(\boldsymbol{u}_i))C_\theta^{-1}(N(\boldsymbol{u}_i))C_\theta(N(\boldsymbol{u}_i), \boldsymbol{u}_i) \end{aligned}$$

From this point, we can define a new covariance function $\tilde{C}_\theta^*$. For any two locations $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ in $\mathcal{D}$, we have

$$\tilde{C}_\theta^*(\boldsymbol{u}_1, \boldsymbol{u}_2) = \begin{cases} \tilde{C}_\theta(\boldsymbol{s}_1, \boldsymbol{s}_2), \text{ if } \boldsymbol{u}_1 = \boldsymbol{s}_1,\ \boldsymbol{u}_2 = \boldsymbol{s}_2 \\ \boldsymbol{B}_{\boldsymbol{u}_1}\tilde{C}_\theta(N(\boldsymbol{u}_1), \boldsymbol{s}_2), \text{ if } \boldsymbol{V}_1 \notin \mathcal{S},\ \boldsymbol{V}_2 = \boldsymbol{s}_2 \\ \boldsymbol{B}_{\boldsymbol{u}_1}\tilde{C}_\theta(N(\boldsymbol{u}_1), N(\boldsymbol{u}_2))\boldsymbol{B}_{\boldsymbol{u}_2}^{'} + 1_{(\boldsymbol{u}_1=\boldsymbol{u}_2)}\boldsymbol{F}_{\boldsymbol{u}_1}, \text{ if } \boldsymbol{V}_1, \boldsymbol{V}_2 \notin \mathcal{S} \end{cases}$$

where $\tilde{C}_\theta$ is the covariance matrix associated with the density of $\tilde{\boldsymbol{w}}_\mathcal{S}$. This completes the construction of the new spatial process which is denoted $NNGP(0, \tilde{C}_\theta^*)$.

The computational advantage lies in the sparsity of the precision matrix obtained under the new covariance function $\tilde{C}_\theta^*$. Assuming that we restrict the size of the neighborhood set to m, the largest matrix inversion required by the MCMC updates are $m \times m$. In fact, the calculations are linear in n ($\mathcal{O}((n+k)m^3)$).

3.1 Divide-and-conquer for NNGP (DICNNGP)

As we have seen in the previous section, the core computational advantage of the NNGP resides in the sparsity of the resulting precision matrix. Such sparsity can be explicitly leveraged in a dimension reduction setting, as illustrated in Banerjee (2017), where the conditional distribution of $\boldsymbol{w}_{\mathcal{U}}$ on $\boldsymbol{w}_{\mathcal{S}}$ (see Equation 2) is rewritten as a linear model:

$$w(\boldsymbol{u}_i) = \sum_{j=1}^{m} \boldsymbol{A}_j(\boldsymbol{u}_i) w(\boldsymbol{s}_j) + \eta(\boldsymbol{u}_i), \;\; \eta(\boldsymbol{u}_i) \sim N(0, \delta^2(\boldsymbol{u}_i)), \tag{4}$$

where $\boldsymbol{A}(\boldsymbol{u}_i) = (a_1(\boldsymbol{u}_i), ..., a_m(\boldsymbol{u}_i))$ and $\delta^2(\boldsymbol{u}_i)$ are fully specified by the covariance function. We can compute the values of $\boldsymbol{A}(\boldsymbol{u}_i)$ and $\delta(\boldsymbol{u}_i)$ efficienctly as:

$$\boldsymbol{A}_{N(\boldsymbol{u}_i)}(\boldsymbol{u}_i) = C_\theta^{-1}(N(\boldsymbol{u}_i)) C_\theta(N(\boldsymbol{u}_i), \boldsymbol{u}_i),$$

and $\forall \boldsymbol{s}_j \notin N(\boldsymbol{u}_i),\ a_j(\boldsymbol{u}_i) = 0$, and

$$\delta^2(\boldsymbol{u}_i) = C_\theta(\boldsymbol{u}_i) - C_\theta(\boldsymbol{u}_i, N(\boldsymbol{u}_i)) \boldsymbol{A}_{N(\boldsymbol{u}_i)}(\boldsymbol{u}_i).$$

This linear representation allows us to use the split-and-merge idea presented in Qian (2018). Consider observations $\boldsymbol{X}, \boldsymbol{Y}$ with n observations and p covariates to be modeled using simple linear regression,

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \;\; \boldsymbol{\epsilon} \sim N_n(0, \sigma^2 I_n)$$

under a conjugate prior $NIG_p(\boldsymbol{\mu}, \boldsymbol{\Lambda}, a, b)$ for $\boldsymbol{\beta}$ and σ^2. The posterior distributions of $\boldsymbol{\beta}$ and σ^2 can be obtained by dividing the data into two blocks $(\boldsymbol{X}_1, \boldsymbol{Y}_1)$ and $(\boldsymbol{X}_2, \boldsymbol{Y}_2)$, computing their respective posterior distributions and merging the results. Denoting the subsamples posterior parameters as $\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i, a_i, b_i$ for $i = 1, 2$, the posterior distribution given the full data is $NIG_p(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Lambda}}, \tilde{a}, \tilde{b})$ with,

$$\begin{aligned}
\tilde{\boldsymbol{\mu}} &= (\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1\boldsymbol{\mu}_1 + \boldsymbol{\Lambda}_2\boldsymbol{\mu}_2), \\
\tilde{\boldsymbol{\Lambda}} &= \boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2, \\
\tilde{a} &= a_1 + a_2 + \frac{p}{2}, \\
\tilde{b} &= b_1 + b_2 + \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu})'(\boldsymbol{\Lambda}_1^{-1})(\boldsymbol{\mu}_1 - \boldsymbol{\mu}) + \frac{1}{2}(\boldsymbol{\mu}_2 - \boldsymbol{\mu})'(\boldsymbol{\Lambda}_2^{-1})(\boldsymbol{\mu}_2 - \boldsymbol{\mu}),
\end{aligned}$$

where p is the dimension of the covariates in the regression model. It is important to highlight that the combined posterior distribution is not an approximation, the resulting inference is exactly what would be obtained by performing the calculations on the complete dataset.

Rewriting Equation (3) for the joint model of $\mathcal{U}$, we have

$$\boldsymbol{w}_{\mathcal{U}} = \boldsymbol{A}\boldsymbol{w}_{\mathcal{S}} + \boldsymbol{\eta}_{\mathcal{U}}, \;\; \boldsymbol{\eta}_{\mathcal{U}} \sim N_n(\boldsymbol{0}, \boldsymbol{D}), \tag{5}$$

where $\boldsymbol{A}$ is the matrix formed by the vectors $\boldsymbol{A}(\boldsymbol{u}_i)$, and $\boldsymbol{D}$ is a diagonal matrix with elements $\delta^2(\boldsymbol{u}_i)$. The joint posterior distribution of $\boldsymbol{w}_{\mathcal{S}}$ and σ^2 given the observations $\boldsymbol{w}_{\mathcal{U}}$ is a $NIG_k(\boldsymbol{\mu}^*, \boldsymbol{V}^*, a^*, b^*)$ where

$$\begin{aligned}
\boldsymbol{\mu}^* &= \boldsymbol{V}^*(\boldsymbol{A}^{'}\boldsymbol{D}^{-1}\boldsymbol{w}_{\mathcal{U}}) \\
\boldsymbol{V}^* &= \left(\boldsymbol{A}^{'}\boldsymbol{D}^{-1}\boldsymbol{A} + \tilde{C}_{\theta}^{-1}(\mathcal{S})\right)^{-1} \\
a^* &= a + \frac{n}{2} \\
b^* &= b + \frac{1}{2}\left(\boldsymbol{w}_{\mathcal{U}}^{'}\boldsymbol{w}_{\mathcal{U}} - \boldsymbol{\mu}^{'*}\boldsymbol{V}^*\boldsymbol{\mu}^*\right)
\end{aligned}$$

where $\tilde{C}_{\theta}^{-1}(\mathcal{S})$ is the sparse prior precision matrix of $\boldsymbol{w}_{\mathcal{S}}$, and a and b are the hyperparameters of the prior on σ^2. Applying the divide-and-conquer algorithm from Qian (2018) with the assumption that we have J blocks, the parameters of the posterior distribution are equivalently computed by

$$\begin{aligned}
\boldsymbol{\mu}^* &= \boldsymbol{V}^* \sum_{j=1}^{J} \boldsymbol{A}_j^{'}\boldsymbol{D}_j^{-1}\boldsymbol{w}_j \\
\boldsymbol{V}^* &= \left(\sum_{j=1}^{J} \boldsymbol{A}_j^{'}\boldsymbol{D}_j^{-1}\boldsymbol{A}_j + \tilde{C}_{\theta}^{-1}(\mathcal{S})\right)^{-1} \\
a^* &= a + \sum_{j=1}^{J} \frac{n_j}{2} \\
b^* &= b + \frac{1}{2}\mu^{'}\tilde{C}_{\theta}^{-1}(\mathcal{S})\mu + \frac{1}{2}\sum_{j=1}^{J}(\boldsymbol{\mu}_j - \boldsymbol{\mu})^{'}(\boldsymbol{\Lambda}_j^{-1})(\boldsymbol{\mu}_j - \boldsymbol{\mu}),
\end{aligned}$$

where

$$\boldsymbol{\mu}_j = \left(\boldsymbol{A}_j' \boldsymbol{D}_j^{-1} \boldsymbol{A}_j\right)' \boldsymbol{A}_j' \boldsymbol{D}_j^{-1} \boldsymbol{w}_j, \boldsymbol{\Lambda}_j = \boldsymbol{A}_j' \boldsymbol{D}_j^{-1} \boldsymbol{A}_j,$$

where $\boldsymbol{A}_j$ is formed by the vectors $\boldsymbol{A}(\boldsymbol{u}_i)$ for $\boldsymbol{u}_i$ in block i, w_j is formed from the observations $w_{\mathcal{U}}$ in block i, and $\boldsymbol{D}_j$ is a diagonal matrix with elements $\delta(\boldsymbol{u}_i)$ for $\boldsymbol{u}_i$ in block j.

A careful analysis of the order of computations highlights the possible gain in efficiency from the parallelization of the model. In the R package spNNGP by Finley et al. (2017), the posterior sampling for the NNGP model is achieved in the best case scenario with computations that are linear in n. The order of the computations also depend on the size of the neighborhood (m^3) and the size of the parameter space (p^3). In the spatial setting that is of interest, we assume that m and p are reasonably small and therefore have a minimal impact on the computational efficiency. In the divide-and-conquer approach suggested, the initial calculations involving the reference grid are linear in k, where k is the size of the reference grid. In an optimal implementation of the algorithm, the posterior inference can be performed with operations that are linear with respect to the size of each server. This implies that the order of calculations is reduced to n/J, where the computations for each block can be done concurrently. Finally, the order of computations rely on the number of neighbors and on the number of parameter in the same way as the implementation of the NNGP. Assuming that θ is held constant, we can review one iteration of the MCMC algorithm to understand the magnitude of the gain.

3.2 Model Development

The implementation of the NNGP model in Datta et al. (2016) suggests to use $\mathcal{U}$ as the reference set. However, in the case where the data is too large to be stored on one computer, this ultimately cannot be achieved. That is, the dimension of the posterior distribution would be the same as the original data, and therefore be too large to be stored. For that reason, the implementation of a distributed approach relies on the use of a reference grid, which is assumed to be known by all servers.

There are two situations that can arise when splitting information on multiple servers.

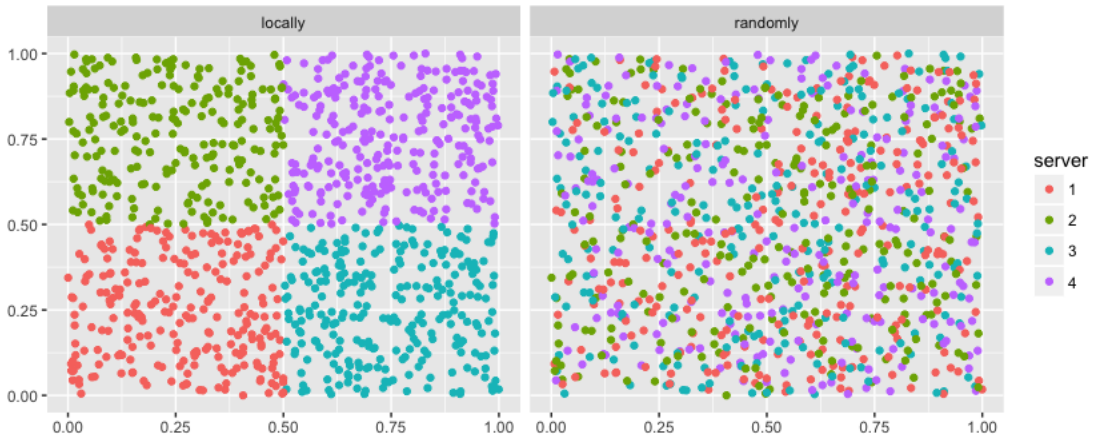


Figure 1: There are two ways to split a database between servers. On the left, the observations are divided locally, and on the right they are split randomly.

As illustrated in Figure 1, the observations can be stored arbitrarily on each server or they can be divided by regions. For instance, if we have two servers, one on the East coast of the United States, and one of the West coast, we can assign observations based on their distance from the servers' locations. The divide-and-conquer approach developed in section 3.1c is independent of the method used to assign observations to servers.

3.2.1 Adding covariates

So far we have been restricting our spatial model to a zero-centered GP. We can expand our model by adding a mean function,

$$y(\boldsymbol{u}_i) = \boldsymbol{X}(\boldsymbol{u}_i)'\boldsymbol{\beta} + w(\boldsymbol{u}_i) + \xi(\boldsymbol{u}_i), \quad \xi(\boldsymbol{u}_i) \sim N(0, \tau^2) \tag{6}$$

where $x(\boldsymbol{u}_i)$ is a vector of p spatially-independent covariates, and $w(\boldsymbol{u}_i)$ is a GP. Since the mean function and the spatial process can be written as a linear model, the joint model for $\mathcal{U} = u_1, ..., u_n$ is available as

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{A} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{w}_{\mathcal{S}} \end{bmatrix} + \boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim N_n(0, \tau^2 I).$$

Assuming the prior for $\boldsymbol{\beta}$ is $N_p(\mathbf{0}, \boldsymbol{K}_\beta)$, the joint posterior distribution for $\boldsymbol{\beta}$ and $\boldsymbol{w}_{\mathcal{S}}$ is $N_{p+k}(\boldsymbol{\mu}^*, \boldsymbol{V}^*)$ where

$$\begin{aligned} \boldsymbol{\mu}^* &= \boldsymbol{V}^* \left(\begin{bmatrix} \boldsymbol{X} & \boldsymbol{A} \end{bmatrix}' \boldsymbol{G}^{-1} \boldsymbol{Y} \right) \\ \boldsymbol{V}^* &= \left(\begin{bmatrix} \boldsymbol{X} & \boldsymbol{A} \end{bmatrix}' \boldsymbol{G}^{-1} \begin{bmatrix} \boldsymbol{X} & \boldsymbol{A} \end{bmatrix} + \tilde{\boldsymbol{K}} \right)^{-1}, \end{aligned}$$

where $\boldsymbol{G} = \boldsymbol{D} + \tau^2 I$, and $\tilde{\boldsymbol{K}} = blockdiag(\tilde{C}_\theta^{-1}(\mathcal{S}), \boldsymbol{K}_\beta)$. Applying the divide-and-conquer construction to these parameters using J blocks, we have,

$$\begin{aligned} \boldsymbol{\mu}^* &= \boldsymbol{V}^* \left(\sum_{j=1}^{J} \begin{bmatrix} \boldsymbol{X}_j & \boldsymbol{A}_j \end{bmatrix}' \boldsymbol{G}_j^{-1} \boldsymbol{Y}_j \right) \\ \boldsymbol{V}^* &= \left(\sum_{j=1}^{J} \begin{bmatrix} \boldsymbol{X}_j & \boldsymbol{A}_j \end{bmatrix}' \boldsymbol{G}_j^{-1} \begin{bmatrix} \boldsymbol{X}_j & \boldsymbol{A}_j \end{bmatrix} + \tilde{\boldsymbol{K}} \right)^{-1}, \end{aligned}$$

where $\boldsymbol{A}_j$, $\boldsymbol{X}_j$, $\boldsymbol{G}_j$ and $\boldsymbol{Y}_j$ include the locations $\boldsymbol{u}_i$ corresponding to block j.

3.2.2 Posterior inference of covariance function parameters

The posterior inference for the nugget τ^2 and range parameter ϕ can be obtained under two different approaches. The first one consists of selecting ϕ and τ^2 based on the maximum marginal likelihood obtained from fitting the model over a grid of possible values. While this method is highly efficient, it only provides a point estimate for the parameters. A more complete approach uses MCMC to sample from the posterior distribution of the two parameters. The drawbacks of this method is that it is computationally expensive as it requires constant communication between the servers and the user node. More specifically, at each iteration, the new parameters must be communicated to the user node, and the covariance matrix must be fully computed for the reference set.

The marginal likelihood in the first scenario can be computed using the outputs from each server. The marginal likelihood distribution is such that

$$m(\boldsymbol{Y}|\theta) = \frac{p(\boldsymbol{Y}|\boldsymbol{\beta}, \sigma^2, \theta) p(\boldsymbol{\beta}, \sigma^2|\theta)}{p(\boldsymbol{\beta}, \sigma^2|\boldsymbol{Y}, \theta)}, \tag{7}$$

holds true for any value of $\boldsymbol{\beta}$. By fixing $\boldsymbol{\beta} = \mathbf{0}$, the sampling distribution simplifies to:

$$\left|\boldsymbol{G}\right|^{-1/2} \exp\left(\frac{1}{2\sigma^2}\boldsymbol{Y}^T\boldsymbol{G}^{-1}\boldsymbol{Y}\right)$$

Finally, since $\boldsymbol{G}$ is diagonal, we can further rewrite the expressions to represent the J servers

$$\begin{aligned}
\boldsymbol{Y}^T\boldsymbol{G}^{-1}\boldsymbol{Y} &= \sum_{i=1}^{n} \frac{\boldsymbol{Y}(\boldsymbol{s}_i)^2}{\delta(\boldsymbol{s}_i)+\tau^2} = \sum_{j=1}^{J}\sum_{i=1}^{n_j} \frac{\boldsymbol{Y}(\boldsymbol{s}_{ji})^2}{\delta(s_{ji})+\tau^2} \\
\left|\boldsymbol{G}\right|^{-1/2} &= \prod_{j=1}^{J} \left|\boldsymbol{G}_j\right|^{-1/2}.
\end{aligned}$$

The calculations of the proposal probability for the MCMC under the second scenario are very similar to the first approach. In this case, the joint posterior distribution for ϕ and τ^2 is:

$$p(\theta|\boldsymbol{\beta}, \sigma^2, \boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{\beta}, \sigma^2, \theta)p(\boldsymbol{\beta}, \sigma^2|\theta).$$

which is equal to the numerator of the marginal likelihood (see Equation 6). Using the simplified calculations outlined above for the sampling distribution, we obtain posterior samples for ϕ and τ^2.

As previously mentioned, while the second method provides more information about the uncertainty of the results, the speed of the computations can be strongly impacted when communication between serves is is an issue.

3.2.3 Posterior predictive distribution

Using the posterior samples for $\boldsymbol{w}_{\mathcal{S}}$, and θ, we can obtain the posterior predictive distribution for any new location $\boldsymbol{u}^*$. Denote $N(\boldsymbol{u}^*)$ as the set of neighbors of $\boldsymbol{u}^*$ in the reference set $\mathcal{S}$. For each posterior sample $\boldsymbol{w}_{\mathcal{S}}^{(b)}, \theta^{(b)}$, $b = 1, ..., B$, we can obtain a posterior predictive sample $\boldsymbol{w}^{(b)}(\boldsymbol{u}^*)$ from

$$\boldsymbol{w}^{(b)}(\boldsymbol{u}^*) = \sum_{j=1}^{m} a_j^{(b)}(\boldsymbol{u}^*) w^{(b)}(\boldsymbol{s}_j)$$

where the linear coefficients are computed from the covariance function as

$$\boldsymbol{A}^{(b)}_{N(\boldsymbol{u}^*)}(\boldsymbol{u}^*) = C^{-1}_{\theta^{(b)}}(N(\boldsymbol{u}^*)) C_{\theta^{(b)}}(N(\boldsymbol{u}^*), \boldsymbol{u}^*),$$

and $a_i^{(b)}(\boldsymbol{u}^*) = 0$ for any location i not in the neighborhood of $\boldsymbol{u}^*$.

4 Applications

In the next section, we apply the divide-and-conquer approach to a simulation as well as to the NASA soil moisture active-passive satellite (SMAP) dataset. The spatial prediction abilities of NNGP have already been established in previous work as seen in Banerjee (2017). The goal of these applications is to illustrate the potential gain in performance by using multiple servers and parallel computing. In addition, we aim to demonstrate the inherent sequential nature of the algorithm. The notion of servers can be extended and applied to the case where blocks of data are recorded sequentially, that is, the methodology proposed in this paper can be used to fit the NNGP and then update the fit as new data become available. This will be discussed further in the context of the analysis of the SMAP dataset, where the servers are defined as the days during which the data were recorded.

4.1 Simulation

Let $\boldsymbol{u}_1, ..., \boldsymbol{u}_{10,000} \in \mathcal{U}$ be randomly generated locations in a unit square, the simulated data was obtained from the following model

$$\boldsymbol{Y}(\boldsymbol{u}_i) = \boldsymbol{X}(\boldsymbol{u}_i)\boldsymbol{\beta} + w(\boldsymbol{u}_i) + \epsilon(\boldsymbol{u}_i), \quad i = 1, ..., 10,000$$

where $w(\boldsymbol{u}_i) \sim GP(0, C_\theta)$, $\epsilon(\boldsymbol{u}_i) \sim N(0, \tau^2)$. Using an exponential covariance function for the spatial process, with range and variance parameters $\phi = 1/3$ and $\sigma^2 = 1$, and observational error variance $\tau^2 = 0.1$. In addition, $p = 6$ independent covariates (X) generated from a zero-centered Normal distribution were included in the model. The associated vector of linear coefficients $\boldsymbol{\beta} = (5, 5, 5, 5, 5, 10)$.

For simplicity, we arbitrarily separated the dataset into two blocks representing two servers. Maximizing the marginal likelihood to estimate ϕ and τ^2, we obtained an interpolated surface for the simulated observations. The results in Figure 2 show the observed values for $\boldsymbol{w}_\mathcal{U}$ on the left, and the interpolated surface obtained for the reference grid on

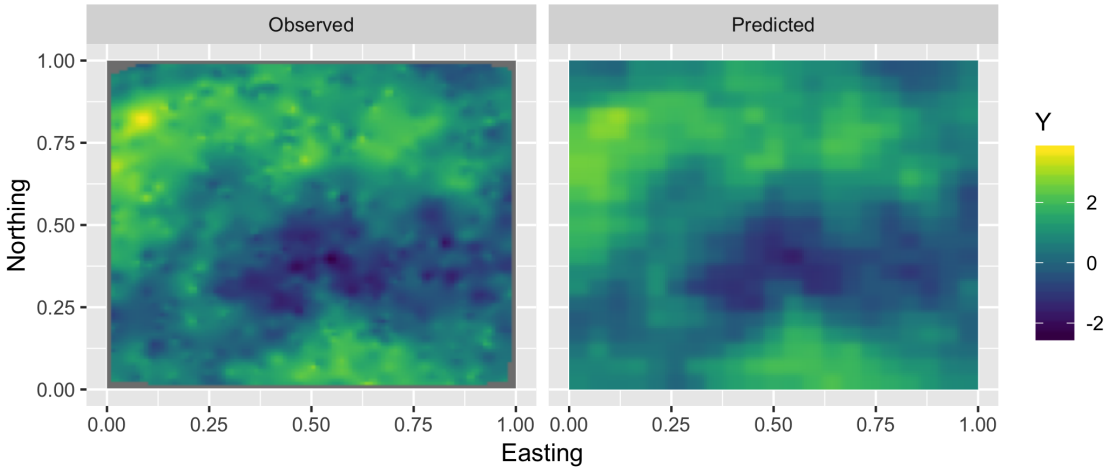


Figure 2: Two surface plots showing the simulated surface (on the left) and the mean posterior predictive surface over the reference grid (on the right).

the right. The predicted values captured the range and the variability of the observations which leads us to conclude that splitting the dataset into blocks did not interfere with the results. Figure 3 shows the marginal likelihood obtained over a grid of values for ϕ and τ^2 with the red cross indicating the parameter values that were selected. In this case, the true value for the range parameter ϕ was recovered but the value of the nugget was slightly underestimated.

Another simulation was used to test the MCMC approach to obtaining posterior inference on ϕ and τ^2. The MCMC algorithm was ran for 5000 iterations with a burn-in period of size 1000. Figure 4 shows the posterior distributions for ϕ and τ^2 along with 95% credible intervals. We notice that the mode is not located at the true value for both parameters which is expected since the NNGP model is an approximation of the true model. Finally, similar to the previous approach, Figure 5 shows the observed values for the spatial process on the left and the predicted values over the reference grid on the right. Again, the range and the local variability of the predictions are in line with fitting the NNGP model directly on the dataset.

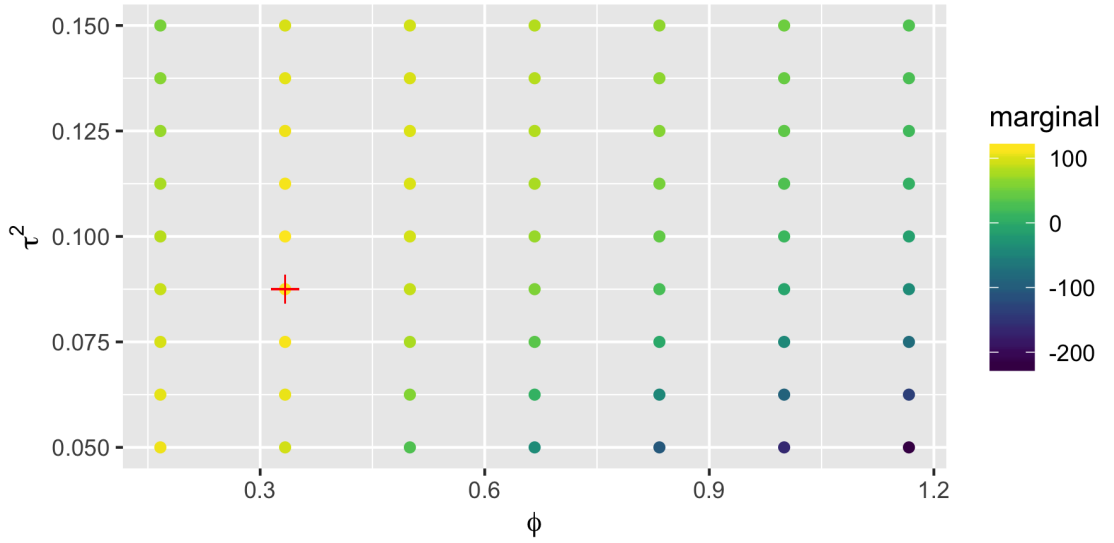


Figure 3: This plot is the marginal likelihood for a grid of possible values for ϕ and τ^2. The blue dot marks the true values, and the red cross marks the values chosen with maximum likelihood. In this simulation, the true values values maximized the marginal likelihood.

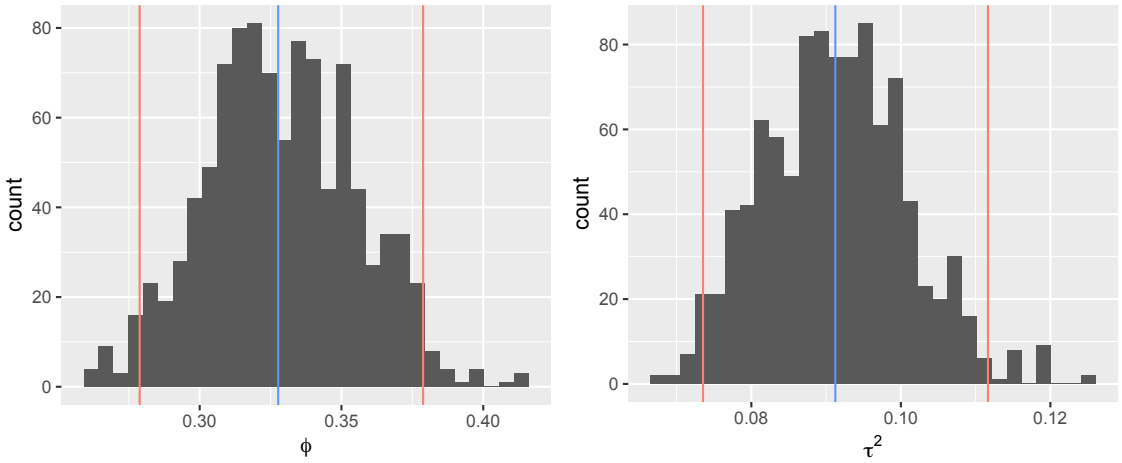


Figure 4: Histograms showing the density of the posterior distributions for ϕ and τ^2.

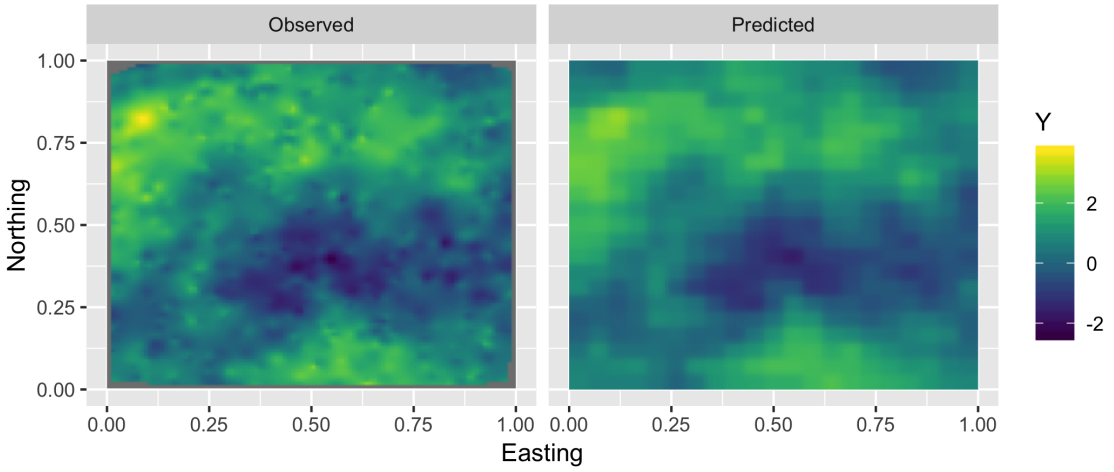


Figure 5: Two surface plots showing the simulated surface and the mean posterior predictive surface obtained.

The objective of the simulations were to show that while the predictions were unchanged by the divide and conquer approach, the runtime for the algorithm can be substantially improved. Figure 6 compares the time elapsed to fit the NNGP using different number of cores to perform the inference. For this simulation, the number of cores corresponded to the number of blocks created from the original dataset. We started with 1 core and gradually increased to 32 cores, which was the maximum number of cores available. The computer used to fit the model was a Dell PE R820 with 4 x Intel Xeon Sandy Bridge E5-4640 processor, each of which has 8 cores per cpu, 2.7 GHz, 16GB RAM, and 1TB SATA hard drive. While the theoretical reduction of the computation is by a factor equivalent to the number of servers J, it is important to consider that a portion of the difference between the times elapsed is due to the matrix multiplication involved in the posterior distribution parameters for the reference grid. In this model, the matrix multiplication has order of operations kn^2. When using J servers, this is reduced to kn^2/J^2, which implies that the calculations are reduced by the squared of the number of servers. It is important to mention that the comparison is not meant as a direct competition between the existing

implementation of the NNGP and the proposed divide-and-conquer method. Rather, the distributed approach presented here is meant as a proof of concept for datasets that are too large to be handled on one computer which would prevent the use of the existing R package.

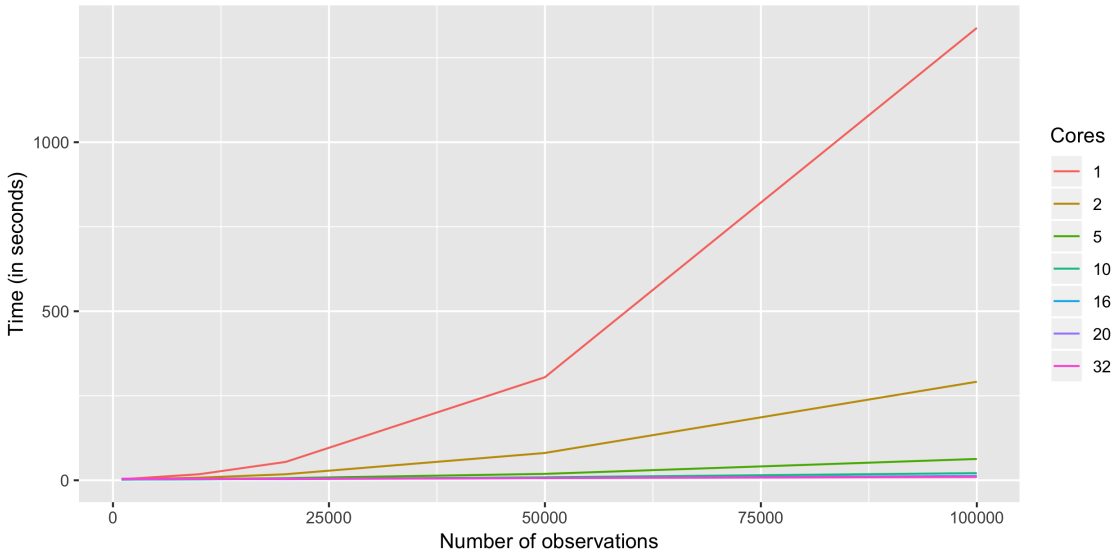


Figure 6: This plot shows the time elapsed to run the simulation based on the number of observations. The green line is the runtime for ten servers, and the red line is the runtime for one server.

4.2 Soil Moisture Data

To illustrate the importance of the distributed computation of likelihood-based spatial models, we applied our algorithm to data collected by the SMAP satellite (NASA, 2018). The SMAP satellite is an Earth satellite that measures soil moisture and freeze state of the top layer of soil as a percentage. Usual applications of soil moisture data range from agricultural productivity to human health. The SMAP mission has been designed to target specifically the understanding of the relationship between soil moisture, the freeze/thaw

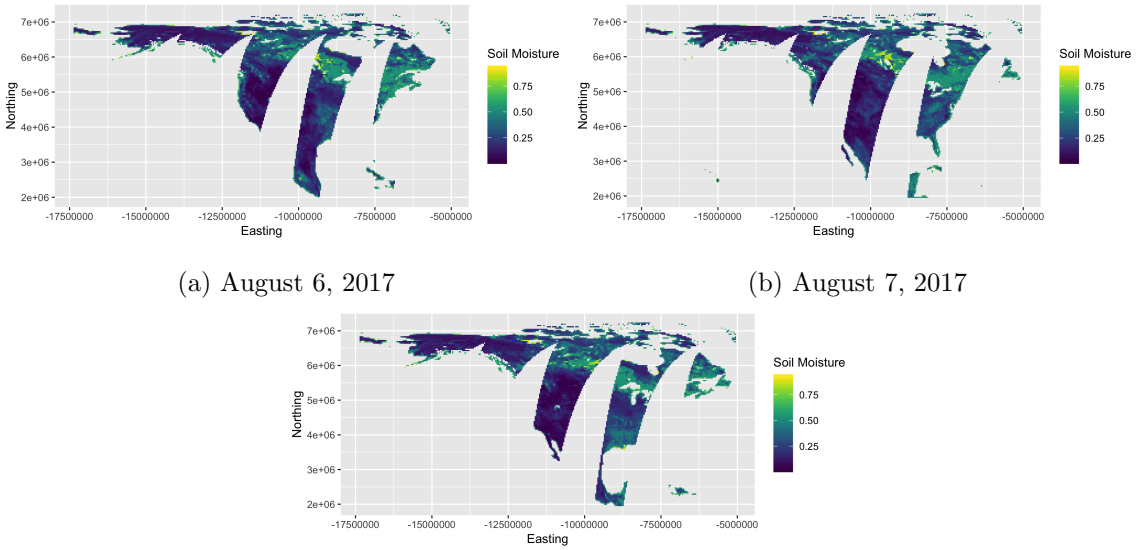
cycle and a variety of environmental constraints. The ultimate goal being to improve weather and climate forecasting (Koster et al., 2018).

The satellite takes on average two days to cover the surface of the Earth. For that reason, we selected three days to conduct our experiment. In this analysis, we wish to demonstrate the sequential feature of the divide-and-conquer algorithm. The GP can be fitted once per day after which the data is no longer needed. This implies that at no point in time one needs to have the data for the three days on one server. This becomes particularly useful for tasks requiring daily updates of massive datasets. As presented in our results, we highlight the evolution of the predictions over the course of the three days of data available from August 6, 2017 to August 8, 2017. The data is retrieved at a resolution of 36km by 36km. By restricting our analysis to North America, the result is a dataset with approximately 10,500 datapoints daily, for a total of 30,000 datapoints. As previously noted, the soil moisture is captured by a percentage for the top layer of soil. In order to transform the range to be applicable for the GP model, we use a probit transformation on the data presented in Figure 7.

As mentioned in section 3.2, a reference grid is needed to accommodate the size of the dataset. To create the reference grid, we first generated a equidistant grid of size 50 by 50 based on a rectangle spanning over North America. We then cropped the grid using a polygon shaped as North America to only keep inland locations. The size of the final reference grid was therefore reduced by more than half and contained 1,173 locations.

The covariance function used in the results presented was the Matern with smoothing parameter $\nu = 3/2$. Different smoothness parameters were investigated but did not lead to significant differences in the resulting predictions. The point estimates for the nugget τ^2 and the covariance range parameter ϕ were obtained after each day by maximizing the marginal likelihood over a grid of possible values. The grid for τ^2 included values between 0.5 and 1.5, and the grid for ϕ ranged from 2.5 and 3.5. For all three days, the pair of parameters that maximized the marginal likelihood were the same at $\tau^2 = 1.5$ and $\phi = 3.5$.

The resulting posterior predictive mean at each reference locations along with the interquartile range are shown in Figures 8 to 10. It is important to highlight that the results



(a) August 6, 2017

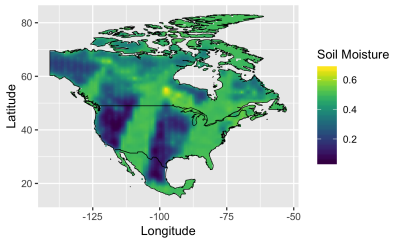
(b) August 7, 2017

(c) August 8, 2017

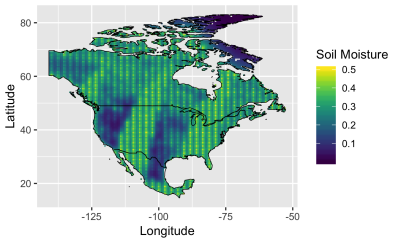
Figure 7: These plots show the soil moisture assessments recovered from the satellite on August 6-8, 2017.

are shown in a different map projection as the initial dataset. The dataset was recorded using a cylindrical equal area projection which made the area of interest not easily recognizable. In order to have a better visual appreciation of the results, the figures are shown using longitude and latitude coordinates. In Figure 11, we show a selection of posterior predictive samples. This allows us to understand and quantify the uncertainty of our predictions in multiple ways. As previously mentioned, we computed the interquartile range of our posterior predictive samples at each reference location. We see that the range is larger for the predictions in Canada. This can be due to the higher variability of the observations in the Canadian region.

For the computations, we split each day into 30 additional blocks, for a total of 90 subsamples. Using the "foreach" function in R over 20 cores, the calculations for one iteration of the marginal likelihood took under 20 seconds. In this scenario also, the computer used to run the inference was a Dell PE R820 with 32 cores.

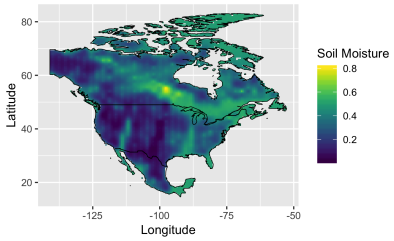


(a) Posterior Predictive Mean

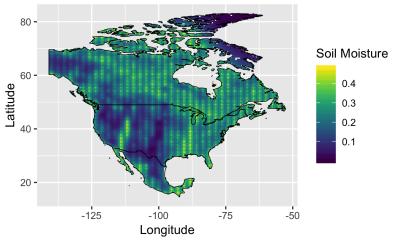


(b) Posterior Predictive Interquartile Range

Figure 8: The plots show the resulting posterior predictions for soil moisture obtained on a grid over North America after performing statistical inference on August 6, 2017.

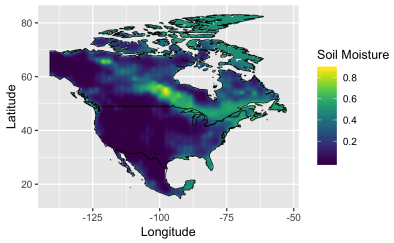


(a) Posterior Predictive Mean

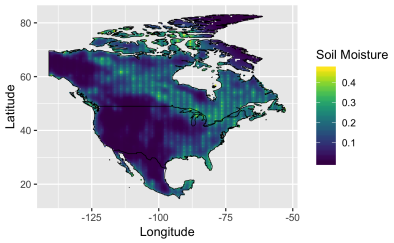


(b) Posterior Predictive Interquartile Range

Figure 9: The plots show the resulting posterior predictions for soil moisture obtained on a grid over North America after performing statistical inference on August 6-7, 2017.



(a) Posterior Predictive Mean



(b) Posterior Predictive Interquartile Range

Figure 10: The plots show the resulting predictions for soil moisture obtained on a grid over North America after performing statistical inference on August 6-8, 2017.

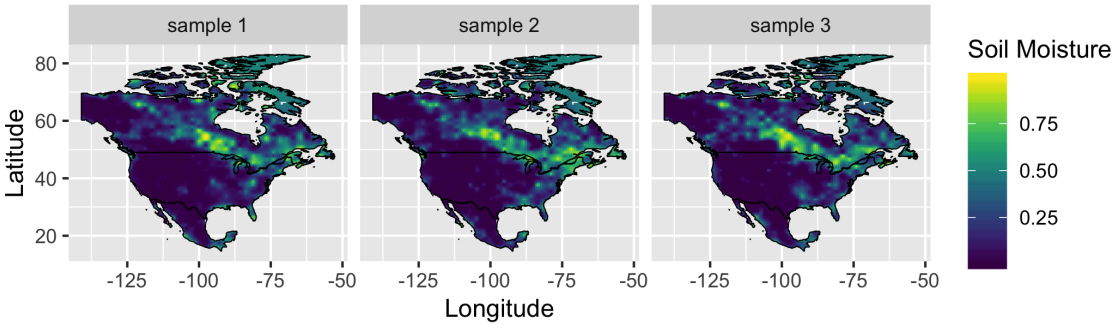


Figure 11: The plots show a selection of three posterior predictive samples for soil moisture obtained on a grid over North America after performing statistical inference on August 6-8, 2017.

5 Conclusions and Future Work

As datasets continue to grow, the need to bring our statistical methods to the data rather than have the data come to us will become a priority. With many spatial applications, the cost of moving data to a central location has become a bottleneck in our ability to perform accurate statistical inference. In this paper, we have laid out a technique to speed up the implementation of NNGP models by distributing the computations by servers. Not only can we improve the computational aspect by a factor proportional to the number of servers available to the user, we also prevent the need to manipulate the data all at once. In that light, we have shown that our method is a good alternative for sequential problems where data is collected over time. The algorithm allows the user to fit model and update the posterior parameters as new data become available.

Finally, while the divide-and-conquer presented focused on the univariate case, the NNGP model has been extended to the multivariate setting. Many applications rely on

the joint modeling of multiple measurements and the implementation of a distributed approach would be largely beneficial.

References

Banerjee, S. (2017). High-dimensional Bayesian Geostatistics. *Bayesian Analysis 12*(2), 583–614.

Banerjee, S., B. Carlin, and A. Gelfand (2014). *Hierarchical Modeling and Analysis of Spatial Data* (second ed.). New York: Chapman and Hall.

Banerjee, S., G. A. E., F. A. O., and S. Huiyan (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70*.

Cressie, N. and G. Johannesson (2008). Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70*(1), 209–226.

Cressie, N. and C. K. Wikle (2011). *Statistics for Spatio-Temporal Data*. Hoboken, NJ: Wiley.

Cressie, N. A. C. (1993). *Statistics for Spatial Data, Revised Edition*. New York: John Wiley and Sons.

Datta, A., S. Banerjee, A. O. Finley, and A. E. Gelfand (2016). Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association 111*(514), 800–812.

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

Furrer, R., M. G. Genton, and D. Nychka (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics 15*(3), 502–523.

Gelfand, A., P. Diggle, M. Fuentes, and P. Guttorp (Eds.) (2010). *Handbook of Spatial Statistics*. Boca Raton, USA: Chapman and Hall.

Guhaniyogi, R., C. Li, T. D. Savitsky, and S. Srivastava (2017, December). A Divide-and-Conquer Bayesian Approach to Large-Scale Kriging. *ArXiv e-prints*.

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

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

Katzfuss, M. and D. Hammerling (2014, 02). Parallel inference for massive distributed spatial data using low-rank models. *Statistics and Computing 27*.

Koster, R. D., Q. Liu, S. P. P. Mahanama, and R. H. Reichle (2018). Improved hydrological simulation using smap data: Relative impacts of model calibration and data assimilation. *Journal of Hydrometeorology 19*(4), 727–741.

NASA (2018). Soil Moisture Active Passive: SMAP_L3_SM_P. *https://smap.jpl.nasa.gov/*.

Qian, H. (2018, 12). Big data Bayesian linear regression and variable selection by normal-inverse-gamma summation. *Bayesian Anal. 13*(4), 1011–1035.

Vecchia, A. V. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society. Series B (Methodological) 50*(2), 297–312.