Nearest Neighbor Gaussian Processes for Large Spatial Data

Andrew Finley[1] & Jeffrey Doser[2]

May 15, 2023

[1] Department of Forestry, Michigan State University.

[2] Department of Integrative Biology, Michigan State University.

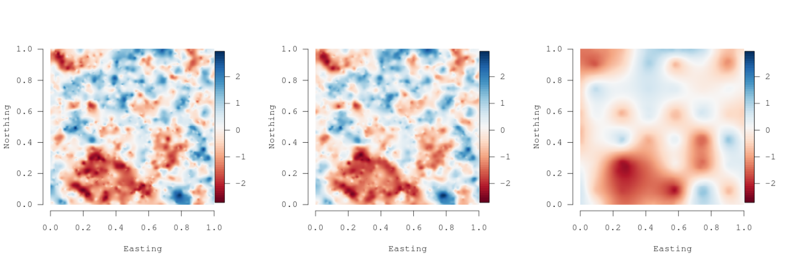
Low rank Gaussian Predictive Process

Pros

- Proper Gaussian process
- Allows for coherent spatial interpolation at arbitrary resolution
- Can be used as prior for spatial random effects in any hierarchical setup for spatial data
- Computationally tractable

Low rank Gaussian Predictive Process

Cons



True *w*

Full GP

PP 64 knots

Figure: Comparing full GP vs low-rank GP with 2500 locations

- Low rank models like the Predictive Process (PP) often tends to oversmooth
- Increasing the number of knots can fix this but will lead to heavy computation

Sparse matrices

- Idea: Use a sparse matrix instead of a low rank matrix to approximate the dense full GP covariance matrix
- Goals:
  - Scalability: Both in terms of storage and computing inverse and determinants
  - Closely approximate full GP inference
  - Proper Gaussian process model like the Predictive Process

Cholesky factors

- Write a joint density $p(w) = p(w_1, w_2, \ldots, w_n)$ as:

$$p(w_1)p(w_2 \,|\, w_1)p(w_3 \,|\, w_1, w_2) \cdots p(w_n \,|\, w_1, w_2, \ldots, w_{n-1})$$

- For Gaussian distribution $w \sim N(0, C)$ this $\Rightarrow$

$$\begin{aligned} w_1 &= 0 + \eta_1; \\ w_2 &= a_{21} w_1 + \eta_2; \\ \ldots & \quad \ldots \quad \ldots \\ w_n &= a_{n1} w_1 + a_{n2} w_2 + \cdots + a_{n,n-1} w_{n-1} + \eta_n; \end{aligned}$$

Cholesky factors

- Write a joint density $p(w) = p(w_1, w_2, \ldots, w_n)$ as:

$$p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_1, w_2) \cdots p(w_n \mid w_1, w_2, \ldots, w_{n-1})$$

- For Gaussian distribution $w \sim N(0, C)$ this $\Rightarrow$

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ a_{21} & 0 & 0 & \ldots & 0 & 0 \\ a_{31} & a_{32} & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \ldots & a_{n,n-1} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} + \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \vdots \\ \eta_n \end{bmatrix}$$

$\Longrightarrow w = Aw + \eta; \quad \eta \sim N(0, D),$ where $D = diag(d_1, d_2, \ldots, d_n)$.

Cholesky factors

- Write a joint density $p(w) = p(w_1, w_2, \ldots, w_n)$ as:

$$p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_1, w_2) \cdots p(w_n \mid w_1, w_2, \ldots, w_{n-1})$$

- For Gaussian distribution $w \sim N(0, C)$ this $\Rightarrow$

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ a_{21} & 0 & 0 & \ldots & 0 & 0 \\ a_{31} & a_{32} & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \ldots & a_{n,n-1} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} + \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \vdots \\ \eta_n \end{bmatrix}$$

$\Longrightarrow w = Aw + \eta; \quad \eta \sim N(0, D),$ where $D = diag(d_1, d_2, \ldots, d_n)$.

- Cholesky factorization: $C^{-1} = (I - A)' D^{-1} (I - A)$

Cholesky factors

- $w_{<i} = (w_1, w_2, \ldots, w_{i-1})'$
- $c_i = Cov(w_i, w_{<i}),\ C_i = Var(w_{<i})$
- i^{th} row of A and $d_i = Var(\eta_i)$ are obtained from $p(w_i \,|\, w_{<i})$ as follows:
    - Solve for a_{ij}'s from $\sum_{j=1}^{i-1} a_{ij} w_j = E(w_i \,|\, w_{<i}) = c_i' C_i^{-1} w_{<i}$

Cholesky factors

- $w_{<i} = (w_1, w_2, \ldots, w_{i-1})'$
- $c_i = Cov(w_i, w_{<i})$, $C_i = Var(w_{<i})$
- i^{th} row of A and $d_i = Var(\eta_i)$ are obtained from $p(w_i \,|\, w_{<i})$ as follows:
    - Solve for a_{ij}'s from $\sum_{j=1}^{i-1} a_{ij} w_j = E(w_i \,|\, w_{<i}) = c_i' C_i^{-1} w_{<i}$
    - $d_i = Var(w_i \,|\, w_{<i}) = \sigma^2 - c_i' C_i^{-1} c_i$

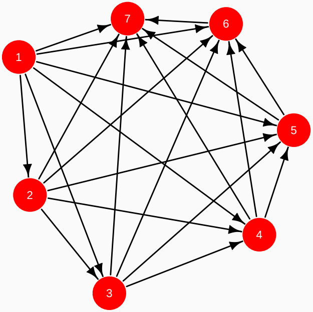
Cholesky factors

- $w_{<i} = (w_1, w_2, \ldots, w_{i-1})'$
- $c_i = Cov(w_i, w_{<i})$, $C_i = Var(w_{<i})$
- i^{th} row of A and $d_i = Var(\eta_i)$ are obtained from $p(w_i \mid w_{<i})$ as follows:
  - Solve for a_{ij}'s from $\sum_{j=1}^{i-1} a_{ij} w_j = E(w_i \mid w_{<i}) = c_i' C_i^{-1} w_{<i}$
  - $d_i = Var(w_i \mid w_{<i}) = \sigma^2 - c_i' C_i^{-1} c_i$

Cholesky factors

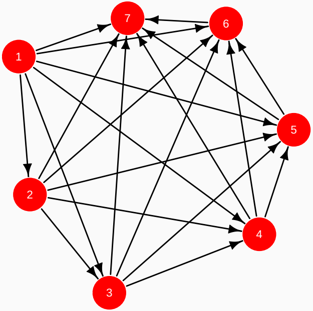
- $w_{<i} = (w_1, w_2, \ldots, w_{i-1})'$
- $c_i = Cov(w_i, w_{<i}),\ C_i = Var(w_{<i})$
- i^{th} row of A and $d_i = Var(\eta_i)$ are obtained from $p(w_i \mid w_{<i})$ as follows:
    - Solve for a_{ij}'s from $\sum_{j=1}^{i-1} a_{ij} w_j = E(w_i \mid w_{<i}) = c_i' C_i^{-1} w_{<i}$
    - $d_i = Var(w_i \mid w_{<i}) = \sigma^2 - c_i' C_i^{-1} c_i$
- For large i, inverting C_i becomes slow
- The Cholesky factor approach for the full GP covariance matrix C does not offer any computational benefits

Cholesky Factors and Directed Acyclic Graphs (DAGs)



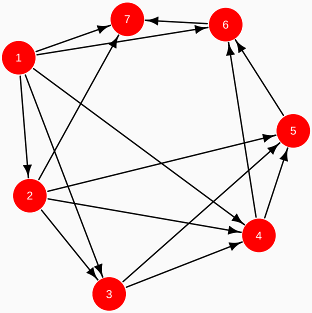
- Number of non-zero entries (sparsity) of A equals number of arrows in the graph
- In particular: Sparsity of the i^{th} row of A is same as the number of arrows towards i in the DAG

Introducing sparsity via graphical models



$$p(y_1)p(y_2 \mid y_1)p(y_3 \mid y_1, y_2)p(y_4 \mid y_1, y_2, y_3)$$
$$\times\ p(y_5 \mid y_1, y_2, y_3, y_4)p(y_6 \mid y_1, y_2, \ldots, y_5)p(y_7 \mid y_1, y_2, \ldots, y_6)\ .$$

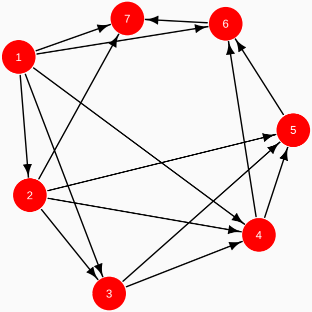
Introducing sparsity via graphical models



$$p(y_1)p(y_2 \mid y_1)p(y_3 \mid y_1, y_2)p(y_4 \mid y_1, y_2, y_3)$$

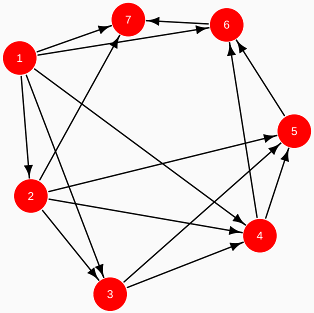
$$p(y_5 \mid \cancel{y_1}, y_2, y_3, y_4)p(y_6 \mid y_1, \cancel{y_2}, \cancel{y_3}, y_4, y_5)p(y_7 \mid y_1, y_2, \cancel{y_3}, \cancel{y_4}, \cancel{y_5}, y_6)$$

Introducing sparsity via graphical models



- Create a sparse DAG by keeping at most m arrows pointing to each node
- Set $a_{ij} = 0$ for all i, j which has no arrow between them
- Fixing $a_{ij} = 0$ introduces conditional independence and w_j drops out from the conditional set in $p(w_i \mid \{w_k : l < i\})$

Introducing sparsity via graphical models



- $N(i)$ denote *neighbor set* of i, i.e., the set of nodes from which there are arrows to i
- $a_{ij} = 0$ for $j \notin N(i)$ and nonzero a_{ij}'s obtained by solving:

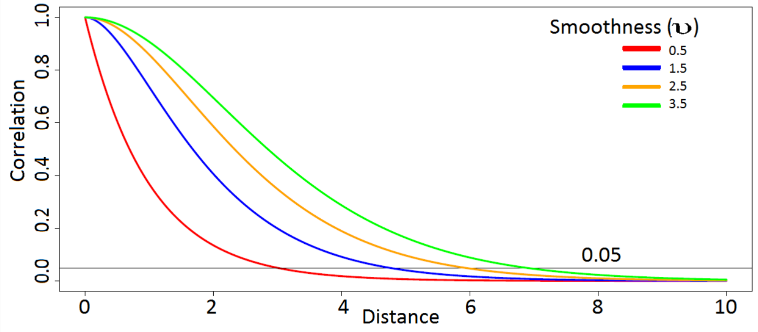
$$\mathrm{E}[w_i \mid w_{N(i)}] = \sum_{j \in N(i)} a_{ij} w_j$$

- The above linear system is only $m \times m$

Choosing neighbor sets

Matern Covariance Function:

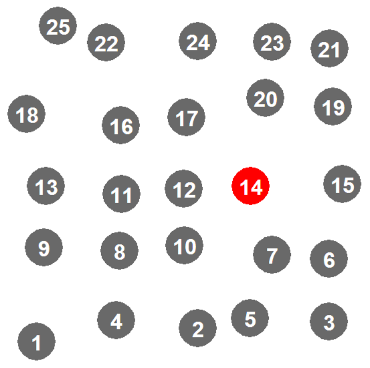
$$C(s_i, s_j) = \frac{1}{2^{\nu-1}\Gamma(\nu)}(||s_i - s_j||\phi)^{\nu}\mathscr{K}_{\nu}(||s_i - s_j||\phi);\ \phi > 0, \nu > 0,$$



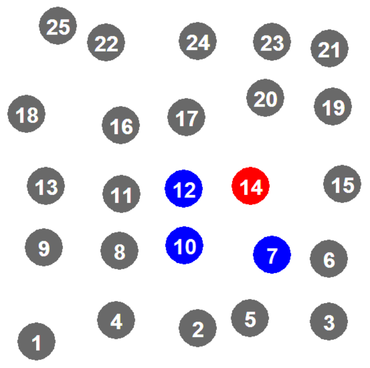
Choosing neighbor sets

- Spatial covariance functions decay with distance
- Vecchia (1988): $N(s_i) = m-$nearest neighbors of s_i in $s_1, s_2, \ldots, s_{i-1}$
  - Nearest points have highest correlations
  - Theory: "Screening effect" – Stein, 2002
- We use Vecchia's choice of m-nearest neighbor
- Other choices proposed in Stein et al. (2004); Gramacy and Apley (2015); Guinness (2016) can also be used

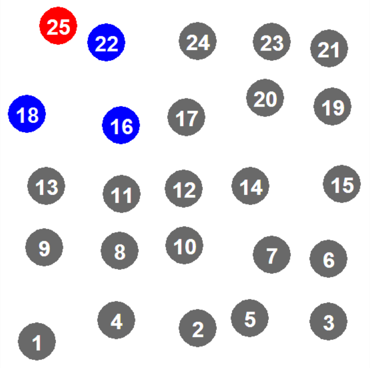
Nearest neighbors



Nearest neighbors

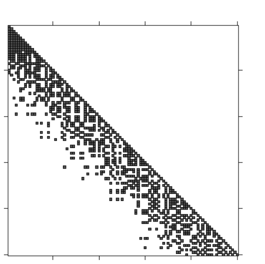


Nearest neighbors

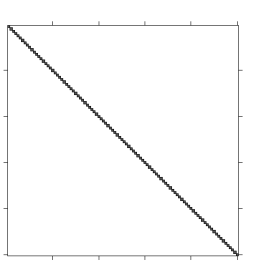


Sparse precision matrices

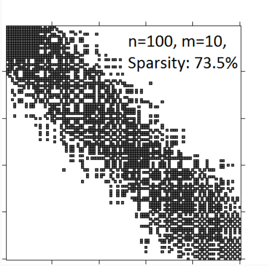
- The neighbor sets and the covariance function $C(\cdot,\cdot)$ define a sparse Cholesky factor A
- $N(w\,|\,0, C) \approx N(w\,|\,0, \tilde{C})\ ;\ \tilde{C}^{-1} = (I - A)^\top D^{-1}(I - A)$



A



D

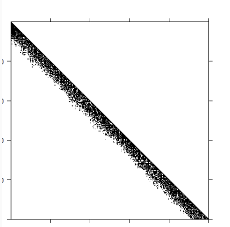


$\tilde{C}^{-1}$

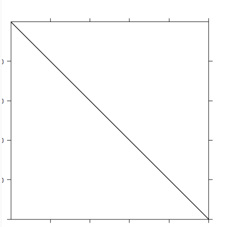
- $\det(\tilde{C}) = \prod_{i=1}^{n} D_i$,
- $\tilde{C}^{-1}$ is sparse with $O(nm^2)$ entries
- Explore some A and $\tilde{C}^{-1}$ sparsity patterns `https://github.com/finleya/NNGP_LDL`

Sparse precision matrices

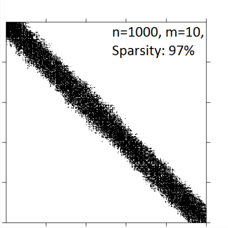
- The neighbor sets and the covariance function $C(\cdot,\cdot)$ define a sparse Cholesky factor A
- $N(w \,|\, 0, C) \approx N(w \,|\, 0, \tilde{C})\ ;\ \tilde{C}^{-1} = (I - A)^\top D^{-1}(I - A)$



A



D



$\tilde{C}^{-1}$

- $\det(\tilde{C}) = \prod_{i=1}^{n} D_i$,
- $\tilde{C}^{-1}$ is sparse with $O(nm^2)$ entries
- Explore some A and $\tilde{C}^{-1}$ sparsity patterns https://github.com/finleya/NNGP_LDL

Extension to a Process

- We have defined $w \sim N(0, \tilde{C})$ over the set of data locations $S = \{s_1, s_2, \ldots, s_n\}$

- For $s \notin S$, define $N(s)$ as set of m-nearest neighbors of s in S

- Define $w(s) = \sum_{i: s_i \in N(s)} a_i(s) w(s_i) + \eta(s)$ where $\eta(s) \overset{ind}{\sim} N(0, d(s))$
  - $a_i(s)$ and $d(s)$ are once again obtained by solving $m \times m$ system

- Well-defined GP over entire domain
  - Nearest Neighbor GP (NNGP) – Datta et al., JASA, (2016)

Hierarchical spatial regression with NNGP

Spatial linear model

$$y(\mathbf{s}) = x(\mathbf{s})'\beta + w(\mathbf{s}) + \epsilon(\mathbf{s})$$

- $w(s)$ modeled as *NNGP* derived from a $GP(0, C(\cdot, \cdot, \,|\, \sigma^2, \phi))$
- $\epsilon(s) \overset{\text{iid}}{\sim} N(0, \tau^2)$ contributes to the nugget
- Priors for the parameters β, σ^2, τ^2 and ϕ
- Only difference from a full GP model is the NNGP prior $w(s)$

Hierarchical spatial regression with NNGP

Full Bayesian Model

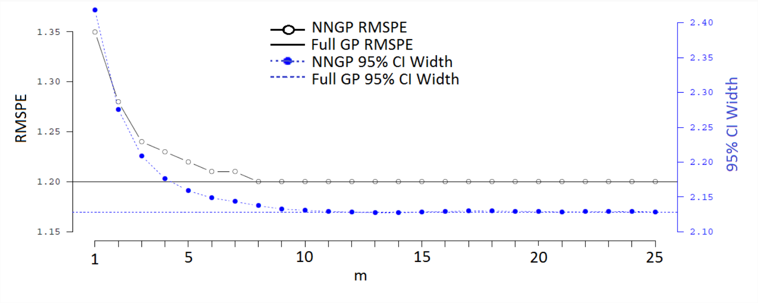
$$N(y \,|\, X\beta + w, \tau^2 I) \times N(w \,|\, 0, \tilde{C}(\sigma^2, \phi)) \times N(\beta \,|\, \mu_\beta, V_\beta)$$
$$\times\, IG(\tau^2 \,|\, a_\tau, b_\tau) \times IG(\sigma^2 \,|\, a_\sigma, b_\sigma) \times Unif(\phi \,|\, a_\phi, b_\phi)$$

Gibbs sampler:

- Conjugate full conditionals for β, τ^2, σ^2 and $w(s_i)$'s
- Metropolis step for updating ϕ
- Posterior predictive distribution at any location using composition sampling:

$$\int N(y(s) \,|\, x(s)'\beta + w(s), \tau^2 I) \times N(w(s) \,|\, a(s)' w_R, d(s)) \times$$
$$p(w, \beta, \tau^2, \sigma^2, \phi \,|\, y)\, d(w, \beta, \tau^2, \sigma^2, \phi)$$

Choosing *m*



- Run NNGP in parallel for few values of *m*
- Choose *m* based on model evaluation metrics
- Our results suggested that typically $m \approx 20$ yielded excellent approximations to the full GP

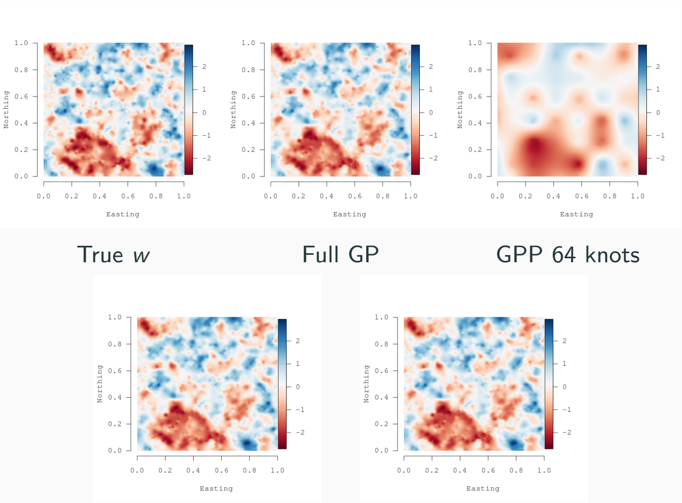
Storage and computation

- Storage:
  - Never needs to store $n \times n$ distance matrix
  - Stores smaller $m \times m$ matrices
  - Total storage requirements $O(nm^2)$
- Computation:
  - Only involves inverting small $m \times m$ matrices
  - Total flop count per iteration of Gibbs sampler is $O(nm^3)$
- Since $m \ll n$, NNGP offers great scalability for large datasets

Simulation experiments

- 2500 locations on a unit square
- $y(s) = \beta_0 + \beta_1 x(s) + w(s) + \epsilon(s)$
- Single covariate $x(s)$ generated as iid $N(0, 1)$
- Spatial effects generated from $GP(0, \sigma^2 R(\cdot, \cdot \,|\, \phi))$
- $R(\cdot, \cdot \,|\, \phi)$ is exponential correlation function with decay ϕ
- Candidate models: Full GP, Gaussian Predictive Process (GPP) with 64 knots and NNGP

Fitted Surfaces



True w Full GP GPP 64 knots

NNGP, $m = 10$ NNGP, $m = 20$

Figure: Univariate synthetic data analysis

Parameter estimates

		NNGP		Predictive Process	Full
	True	$m = 10$	$m = 20$	64 knots	Gaussian Process
β_0	1	1.00 (0.62, 1.31)	1.03 (0.65, 1.34)	1.30 (0.54, 2.03)	1.03 (0.69, 1.34)
β_1	5	5.01 (4.99, 5.03)	5.01 (4.99, 5.03)	5.03 (4.99, 5.06)	5.01 (4.99, 5.03)
σ^2	1	0.96 (0.78, 1.23)	0.94 (0.77, 1.20)	1.29 (0.96, 2.00)	0.94 (0.76, 1.23)
τ^2	0.1	0.10 (0.08, 0.13)	0.10 (0.08, 0.13)	0.08 (0.04, 0.13)	0.10 (0.08, 0.12)
ϕ	12	12.93 (9.70, 16.77)	13.36 (9.99, 17.15)	5.61 (3.48, 8.09)	13.52 (9.92, 17.50)

Model evaluation

	NNGP		Predictive Process	Full
	$m = 10$	$m = 20$	64 knots	Gaussian Process
DIC score	2390	2377	13678	2364
RMSPE	1.2	1.2	1.68	1.2
Run time (Minutes)	14.40	46.47	43.36	560.31

- NNGP performs at par with Full GP
- GPP oversmooths and performs much worse both in terms of parameter estimation and model comparison
- NNGP yields huge computational gains

Spatially varying coefficients

- Often the relationship between the (univariate) spatial response and covariates vary across the space
- The regression coefficients can then be modeled as spatial processes
- Spatially varying coefficient (SVC) model: $y(s) = x(s)'\beta(s) + \epsilon(s)$
- Even though the response can be univariate, $\beta(s)$ is modeled as a p-variate GP

Multivariate GPs

- $Cov(w(s_i), w(s_j)) = C(s_i, s_j \,|\, \theta)$ – a $q \times q$ cross-covariance matrix
- Choices for the function $C(\cdot, \cdot \,|\, \theta)$
  - Multivariate Matérn
  - Linear model of co-regionalization
- For data observed at n locations, all choices lead to a dense $nq \times nq$ matrix $C = Cov(w(s_1), w(s_2), \ldots, w(s_n))$
- Not scalable when nq is large

Multivariate NNGPs

- Cholesky factor approach similar to the univariate case

$$\begin{bmatrix} w(s_1) \\ w(s_2) \\ w(s_3) \\ \vdots \\ w(s_n) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ A_{21} & 0 & 0 & \dots & 0 & 0 \\ A_{31} & A_{32} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{n1} & A_{n2} & A_{n3} & \dots & A_{n,n-1} & 0 \end{bmatrix} \begin{bmatrix} w(s_1) \\ w(s_2) \\ w(s_3) \\ \vdots \\ w(s_n) \end{bmatrix} + \begin{bmatrix} \eta(s_1) \\ \eta(s_2) \\ \eta(s_3) \\ \vdots \\ \eta(s_n) \end{bmatrix}$$

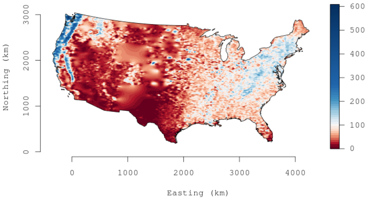
$$\implies w = Aw + \eta; \quad \eta \sim N(0, D), \; D = diag(D_1, D_2, \dots, D_n).$$

- Only differences: $w(s_i)$ and $\eta(s_i)$'s are $q \times 1$ vectors and A_{ij} and D_i's are $q \times q$ matrix

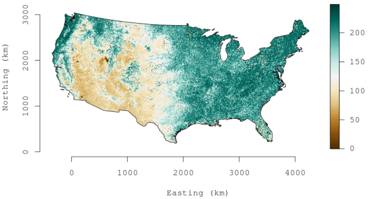
Multivariate NNGPs

- Choose neighbor sets $N(i)$ for each location s_i
- Set $A_{ij} = 0$ if $j \notin N(i)$
- Solve for non-zero A_{ij}'s from the $mq \times mq$ linear system: $\sum_{j \in N(i)} A_{ij} w(s_j) = E(w(s_i) \,|\, \{w(s_j) \,|\, j \in N(i)\})$
- Multivariate NNGP: $w \sim N(0, \tilde{C})$ where $\tilde{C}^{-1} = (I - A)' D^{-1} (I - A)$
- $\tilde{C}^{-1}$ is sparse with $O(nm^2)$ non-zero $q \times q$ blocks
- $\det(\tilde{C}) = \prod_{i=1}^{n} \det(D_i)$
- Storage and computation needs remains linear in n

U.S. Forest biomass data



Observed biomass



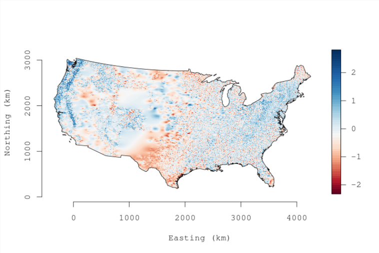
NDVI

- Forest biomass data from measurements at 114,371 plots
- NDVI (greenness) is used to predict forest biomass

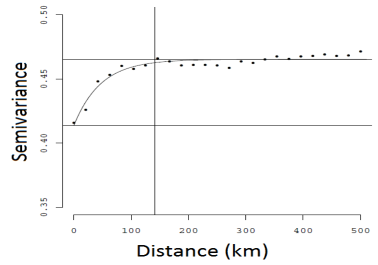
U.S. Forest biomass data

Non Spatial Model

$Biomass = \beta_0 + \beta_1 NDVI + error, \quad \hat{\beta}_0 = 1.043, \; \hat{\beta}_1 = 0.0093$



Residuals



Variogram of residuals

Strong spatial pattern among residuals

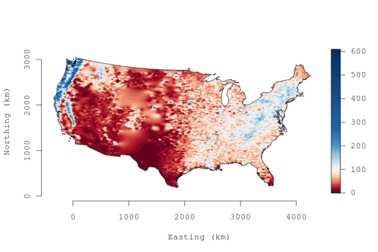
Forest biomass dataset

- $n \approx 10^5$ (Forest Biomass) $\Rightarrow$ full GP requires storage $\approx 40 Gb$ and time $\approx$ 140 hrs per iteration.
- We use a spatially varying coefficients NNGP model

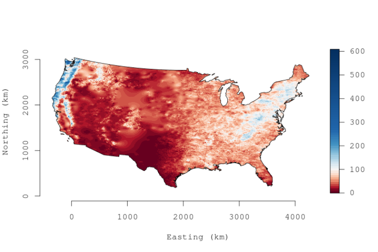
Model

- $Biomass(s) = \beta_0(s) + \beta_1(s) NDVI(s) + \epsilon(s)$
- $w(s) = (\beta_0(s), \beta_1(s))^\top \sim$ Bivariate $NNGP(0, \tilde{C}(\cdot, \cdot \mid \theta))$, $m = 5$
- Time $\approx$ 6 seconds per iteration
- Full inferential output: 41 hours (25000 MCMC iterations)

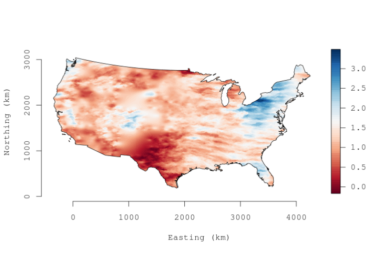
Forest biomass data



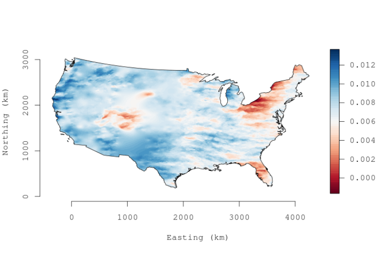
Observed biomass



Fitted biomass



$\beta_0(s)$



$\beta_{NDVI}(s)$

Reducing parameter dimensionality

- The Gibbs sampler algorithm for the NNGP updates $w(s_1), w(s_2), \ldots, w(s_n)$ sequentially
- Dimension of the MCMC for this sequential algorithm is $O(n)$
- If the number of data locations n is very large, this high-dimensional MCMC can converge slowly
- Although each iteration for the NNGP model will be very fast, many more MCMC iterations may be required

Collapsed NNGP

- Same model:

$$
\begin{aligned}
y(s) &= x(s)'\beta + w(s) + \epsilon(s) \\
w(s) &\sim NNGP(0, C(\cdot, \cdot \,|\, \theta)) \\
\epsilon(s) &\overset{\text{iid}}{\sim} N(0, \tau^2)
\end{aligned}
$$

- Vector form $y \sim N(X\beta + w, \tau^2 I); w \sim N(0, \tilde{C}(\theta))$
- Collapsed model: Marginalizing out w, we have $y \sim N(X\beta, \tau^2 I + \tilde{C}(\theta))$

Collapsed NNGP

Model

$y \sim N(X\beta, \tau^2 I + \tilde{C}(\theta))$

- Only involves few parameters β, τ^2 and $\theta = (\sigma^2, \phi)'$
- Drastically reduces the MCMC dimensionality
- Gibbs sampler updates are based on sparse linear systems using $\tilde{C}^{-1}$
- Improved MCMC convergence
- Can recover posterior distribution of $w \mid y$
- Complexity of the algorithm depends on the design of the data locations and is not guaranteed to be $O(n)$

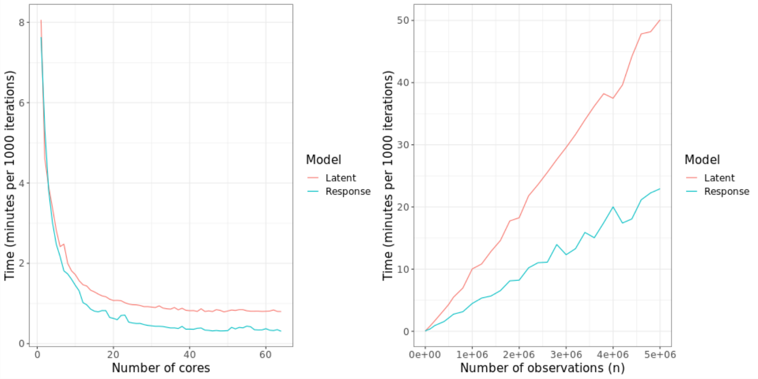
Response NNGP

- $w(s) \sim GP(0, C(\cdot, \cdot \,|\, \theta)) \Rightarrow y(s) \sim GP(x(s)'\beta, \Sigma(\cdot, \cdot \,|\, \tau^2, \theta))$
- $\Sigma(s_i, s_j) = C(s_i, s_j \,|\, \theta) + \tau^2\, \delta(s_i = s_j)$ (δ is Kronecker delta)
- We can directly derive the NNGP covariance function corresponding to $\Sigma(\cdot, \cdot)$
- $\tilde{\Sigma}$ is the NNGP covariance matrix for the n locations
- Response model: $y \sim N(X\beta, \tilde{\Sigma})$
- Storage and computations are guaranteed to be $O(n)$
- Low dimensional MCMC ⇒ Improved convergence
- Cannot coherently recover $w \,|\, y$

Comparison of NNGP models

	Latent	Collapsed	Response
$O(n)$ time	Yes	No	Yes
Recovery of $w \mid y$	Yes	Yes	No
Parameter dimensionality	High	Low	Low

Comparison of NNGP models



b

a

Figure: (a) Runtime for 1000 MCMC iterations for $n = 100000$ and different number of cores. (b) Runtime for 1000 MCMC iterations using 40 cores and n from 1000 to 5 million. Model type (latent and response) refers to different NNGP parameterizations.

Summary of Nearest Neighbor Gaussian Processes

- Sparsity inducing Gaussian process
- Constructed from sparse Cholesky factors based on m nearest neighbors
- Scalability: Storage, inverse and determinant of NNGP covariance matrix are all $O(n)$
- Proper Gaussian process, allows for inference using hierarchical spatial models and predictions at arbitrary spatial resolution
- Closely approximates full GP inference, does not oversmooth like low rank models
- Extension to multivariate NNGP
- Collapsed and response NNGP models with improved MCMC convergence
- spNNGP package in R for analyzing large spatial data using NNGP models