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.

Consider again the spatially-varying intercept model for generic location $\mathbf{s}$

$$y(\mathbf{s}) = \mathbf{x}(\mathbf{s})^\top \boldsymbol{\beta} + w(\mathbf{s}) + \epsilon(\mathbf{s}), \quad \mathbf{s} \in \mathscr{D} \subseteq \mathbb{R}^d,$$

where

$y(\mathbf{s})$ is the outcome,

$\mathbf{x}(\mathbf{s})$ is $p \times 1$ set of predictors including an intercept,

$\boldsymbol{\beta}$ is a vector of p regression parameters,

$w(\mathbf{s})$ is a spatial random effect,

$\epsilon(\mathbf{s})$ is the independent noise process with variance τ^2.

Likelihood from (full rank) GP models

- Assuming $w(\mathbf{s}) \sim GP(0, K_\theta(\cdot,\cdot))$ implies that for a set of n locations[1]

$$\mathbf{w} = (w(\mathbf{s}_1), w(\mathbf{s}_2), \ldots, w(\mathbf{s}_n))^\top \sim MVN(\mathbf{0}, \mathbf{K}_\theta)$$

- Estimating process parameters from the likelihood involves:

$$p(\mathbf{w}) \propto -\frac{1}{2}\log\boxed{\det(\mathbf{K}_\theta)} - \frac{1}{2}\mathbf{w}^\top\boxed{\mathbf{K}_\theta^{-1}}\mathbf{w}$$

- Bayesian inference: priors on $\boldsymbol{\theta}$ and many Markov chain Monte Carlo (MCMC) iterations

[1] $K_\theta(\cdot,\cdot)$ is any valid spatial covariance function, e.g., $\sigma^2 R(\cdot,\cdot;\phi)$, with $\boldsymbol{\theta} = \left(\sigma^2, \phi\right)$.

Computation issues

- Storage: n^2 pairwise distances to compute $\mathbf{K}_\theta$
- $\mathbf{K}_\theta$ is dense; Need to solve $\mathbf{K}_\theta \mathbf{x} = \mathbf{b}$ and need $\det(\mathbf{K}_\theta)$
- This is best achieved using $\texttt{chol}(\mathbf{K}_\theta) = \mathbf{L}\mathbf{D}\mathbf{L}^\top$
- Complexity: roughly $O(n^3)$ flops

Computationally infeasible for large datasets

Burgeoning literature on spatial big data

- Low-rank models: (Wahba, 1990; Higdon, 2002; Rasmussen and Williams, 2006; Cressie and Johannesson, 2008; Banerjee et al., 2008, 2010; Gramacy and Lee, 2008; Finley et al., 2009; Lemos and Sansó, 2009; Sang et al., 2011; Sang and Huang, 2012; Guhaniyogi et al., 2011; Katzfuss and Hammerling, 2017)
- Spectral approximations and composite likelihoods: (Fuentes, 2007; Paciorek, 2007; Eidsvik et al., 2014)
- Multi-resolution approaches: (Nychka et al., 2015; Johannesson et al., 2007; Katzfuss, 2017; Guhaniyogi and Sanso, 2020)
- Sparsity: (Solve $\mathbf{Ax} = \mathbf{b}$ by (i) sparse $\mathbf{A}$, or (ii) sparse $\mathbf{A}^{-1}$)
  1. Covariance tapering (Furrer et al., 2006; Du et al., 2009; Kaufman et al., 2008; Stein, 2013; Shaby and Ruppert, 2012)
  2. GMRFs to GPs: `INLA` (Rue et al., 2009; Lindgren et al., 2011)
  3. LAGP Gramacy et al., 2014; Gramacy and Apley, 2015)
  4. Nearest-neighbor Gaussian Process (NNGP) models (Datta et al., 2016a,c,b; Finley et al., 2019a) builds on Vecchia (1988).

Reduced (Low) rank models

- $\mathbf{K}_\theta \approx \mathbf{J}_\theta \mathbf{K}^*_\theta \mathbf{J}^\top_\theta + \mathbf{D}_\theta$
- $\mathbf{J}_\theta$ is $n \times r$ matrix of spatial basis functions, $r << n$
- $\mathbf{K}^*_\theta$ is $r \times r$ spatial covariance matrix
- $\mathbf{D}_\theta$ is either diagonal or sparse
- Examples: Kernel projections, Splines, Predictive process, FRK, spectral basis ...
- Computations exploit above structure: roughly $O(nr^2) << O(n^3)$ flops

Reduced (Low) rank models (cont'd)

Low-rank models: hierarchical approach

$$N(\mathbf{w}^* \,|\, \mathbf{0}, \mathbf{K}_\theta^*) \times N(\mathbf{w} \,|\, \mathbf{J}_\theta \mathbf{w}^*, \mathbf{D})$$

- $\mathbf{w}$ is $n \times 1$ and n is large
- $\mathbf{w}^*$ is $r \times 1$, where $r << n$, defined over a user-defined set of locations, or knots, $\mathscr{S}^* = \{\mathbf{s}_1^*, \mathbf{s}_2^*, \ldots, \mathbf{s}_r^*\}$.
- $\mathbf{J}_\theta$ is $n \times r$ is a matrix of "basis" functions
- $\mathbf{D}$ is $n \times n$, but easy to invert (e.g., diagonal)
- Derive var($\mathbf{w}$) (or var($\mathbf{w}^* \,|\, \mathbf{y}$)) in alternate ways to obtain

  $$(\mathbf{J}_\theta \mathbf{K}_\theta^* \mathbf{J}_\theta^\top + \mathbf{D})^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1}\mathbf{J}_\theta(\mathbf{K}_\theta^{*-1} + \mathbf{J}_\theta^\top \mathbf{D}^{-1}\mathbf{J}_\theta)^{-1}\mathbf{J}_\theta^\top \mathbf{D}^{-1} \,.$$

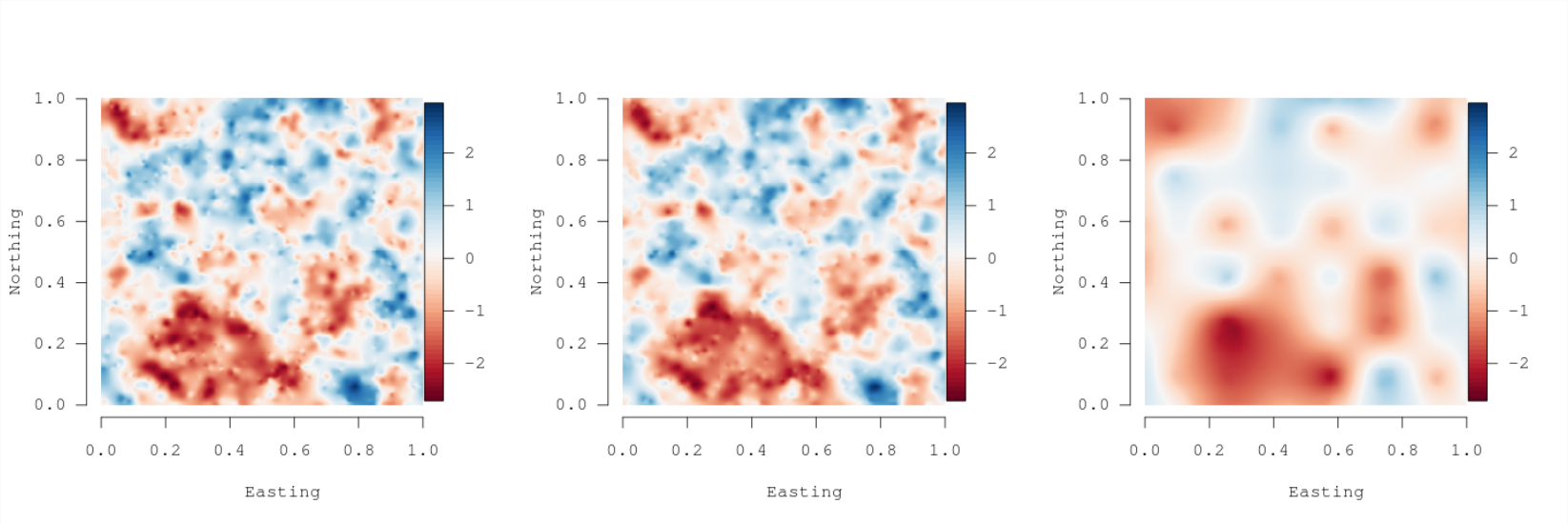
  This is the famous Sherman-Woodbury-Morrison formula.

See, e.g., Finley et al. (2017) for implantation details and software for the Gaussian predictive process (GPP) model.

Simulation experiment

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

Oversmoothing due to reduced-rank models



True w Full GP GPP 64 knots

Figure: Comparing full GP vs low-rank GPP with 2500 locations. Figure (c) exhibits oversmoothing by a low-rank process (predictive process with 64 knots)

See Stein (2014) for very good reasons NOT to use reduced-rank spatial models.

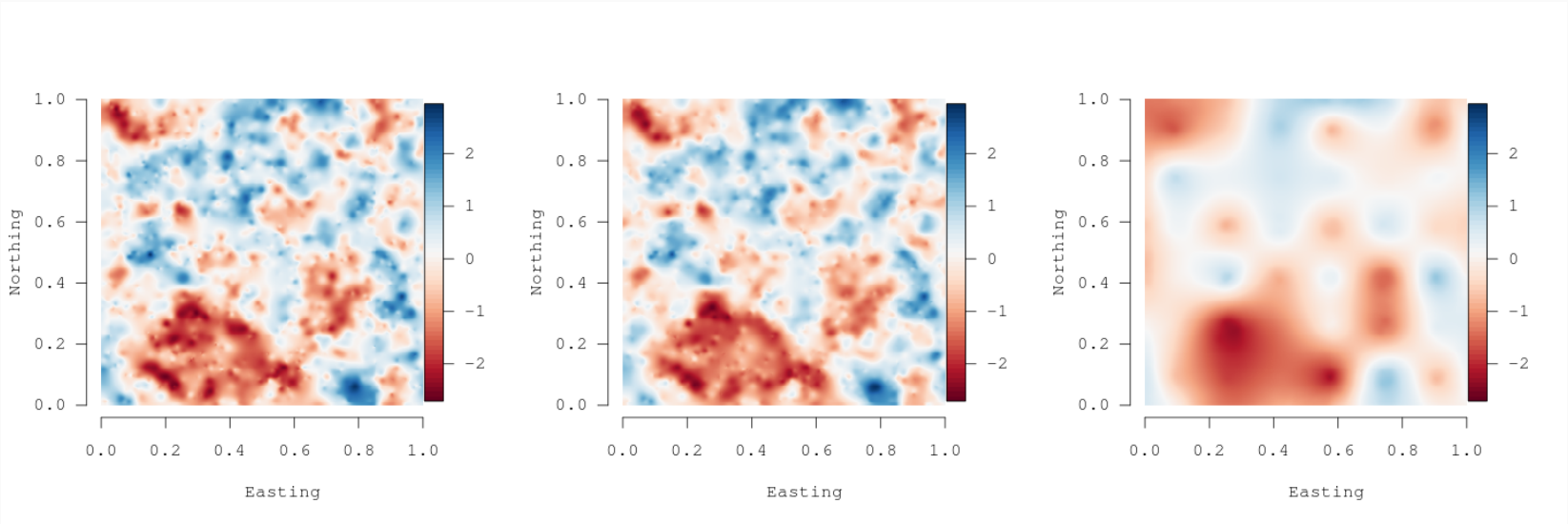
Low rank Gaussian Predictive Process

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 GPP, tend 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 determinant
  - Closely approximate full GP inference
  - Proper Gaussian process model like the GPP

Cholesky factors

- Write a joint density $p(\mathbf{w}) = p(w_1, w_2, \ldots, w_n)$ 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 $\mathbf{w} \sim N(\mathbf{0}, \mathbf{K}_\theta)$ 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 $p(\mathbf{w}) = p(w_1, w_2, \ldots, w_n)$ 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 $\mathbf{w} \sim N(\mathbf{0}, \mathbf{K}_\theta)$ this $\Rightarrow$

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

$$\Longrightarrow \mathbf{w} = \mathbf{A}\mathbf{w} + \boldsymbol{\eta}; \quad \boldsymbol{\eta} \sim N(\mathbf{0}, \mathbf{D}).$$

Cholesky factors

- Write a joint density $p(\mathbf{w}) = p(w_1, w_2, \ldots, w_n)$ 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 $\mathbf{w} \sim N(\mathbf{0}, \mathbf{K}_\theta)$ this $\Rightarrow$

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

$$\Longrightarrow \mathbf{w} = \mathbf{A}\mathbf{w} + \boldsymbol{\eta}; \quad \boldsymbol{\eta} \sim N(\mathbf{0}, \mathbf{D}).$$

- Cholesky factorization:

$$\mathbf{K}_\theta = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{D}(\mathbf{I} - \mathbf{A})^{-\top}, \text{ where } \mathbf{D} = \text{diag}(\text{var}\{w_i \mid w_{\{j<i\}}\})$$

Cholesky factors

- For Gaussian distribution $N(\mathbf{w} \mid \mathbf{0}, \mathbf{K}_\theta)$,

$$\mathbf{K}_\theta = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{D}(\mathbf{I} - \mathbf{A})^{-\top}; \quad \mathbf{D} = \text{diag}(\text{var}\{w_i \mid w_{\{j<i\}}\})$$

- If $\mathbf{L}$ is from $\texttt{chol}(\mathbf{K}_\theta) = \mathbf{L}\mathbf{D}\mathbf{L}^\top$, then $\mathbf{L}^{-1} = \mathbf{I} - \mathbf{A}$.

- a_{ij}'s obtained from $n-1$ linear systems by comparing coefficients of w_j's in

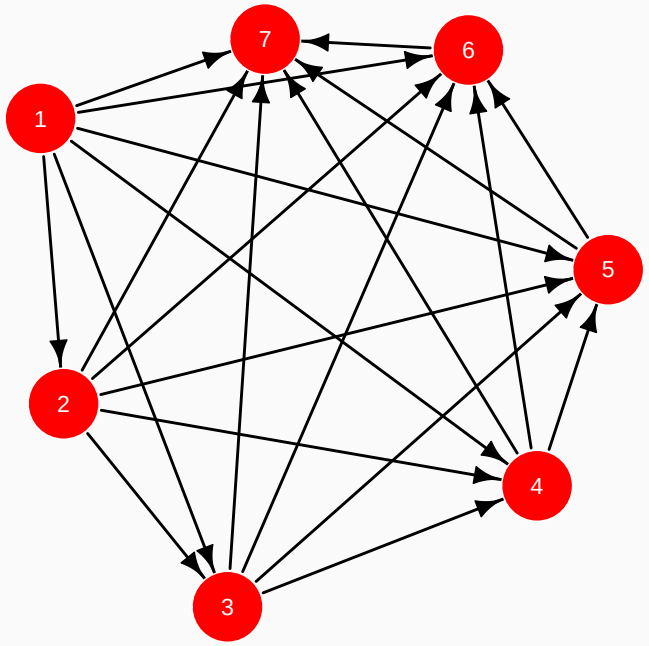
$$\sum_{j<i} a_{ij} w_j = \mathrm{E}[w_i \mid w_{\{j<i\}}] \quad i = 2, \ldots, n$$

- Non-zero elements of $\mathbf{A}$ and $\mathbf{D}$ are computed:

`D[1,1] = K[1,1]` and first row of `A` is zero.

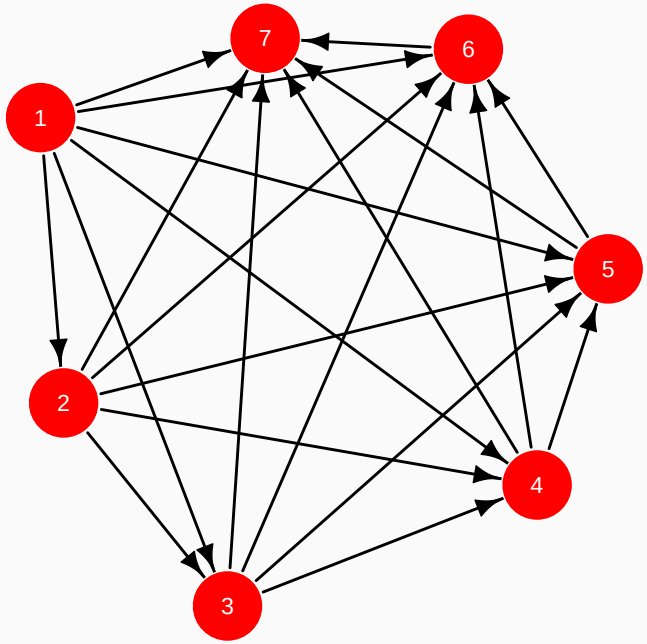
```
for(i in 1:(n-1)) {
  A[i+1,1:i] = solve(K[1:i,1:i], K[1:i,i+1])
  D[i+1,i+1] = K[i+1,i+1] - dot(K[i+1,1:i],A[i+1,1:i])
}
```

Cholesky Factors and Directed Acyclic Graphs (DAGs)



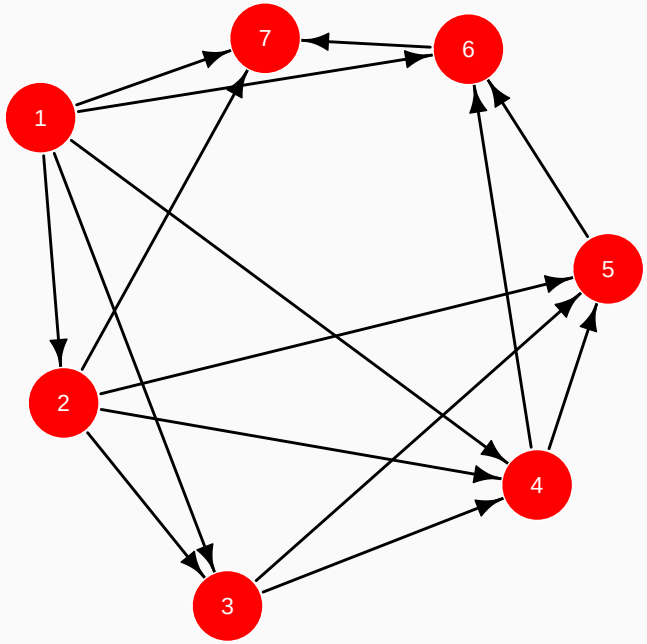
- Number of non-zero entries (sparsity) of **A** equals number of arrows in the graph
- In particular: Sparsity of the i^{th} row of **A** is same as the number of arrows towards i in the DAG

Introducing sparsity via graphical models



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

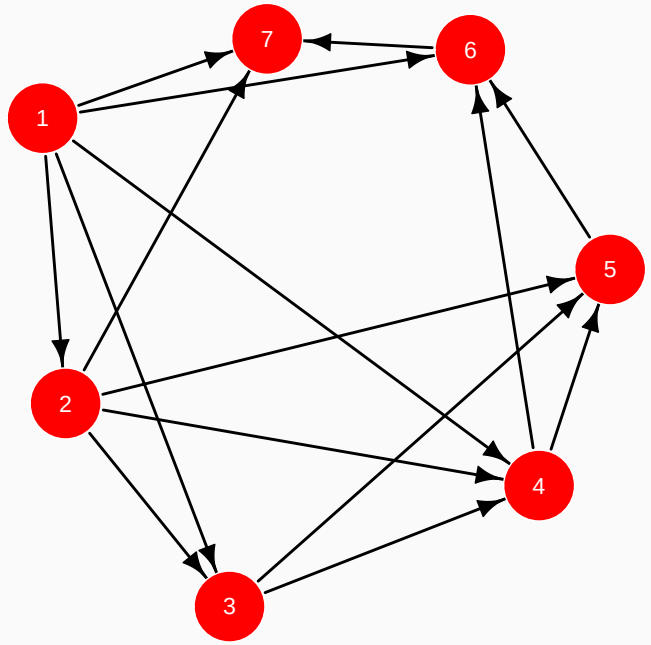
Introducing sparsity via graphical models



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

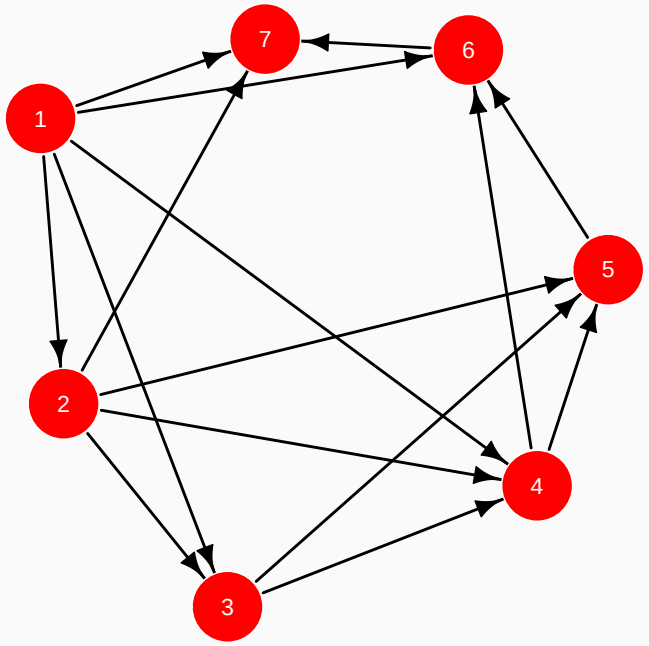
$$p(y_5 \mid \cancel{y_1}, y_2, y_3, y_4)p(y_6 \mid y_1, \cancel{y_2}, \cancel{y_3}, y_4, y_5)p(y_7 \mid y_1, y_2, \cancel{y_3}, \cancel{y_4}, \cancel{y_5}, y_6)$$

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 $p(w_i \mid \{w_k : k < i\})$

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 $m \times m$

- Non-zero elements of sparse **A** and **D** are computed:

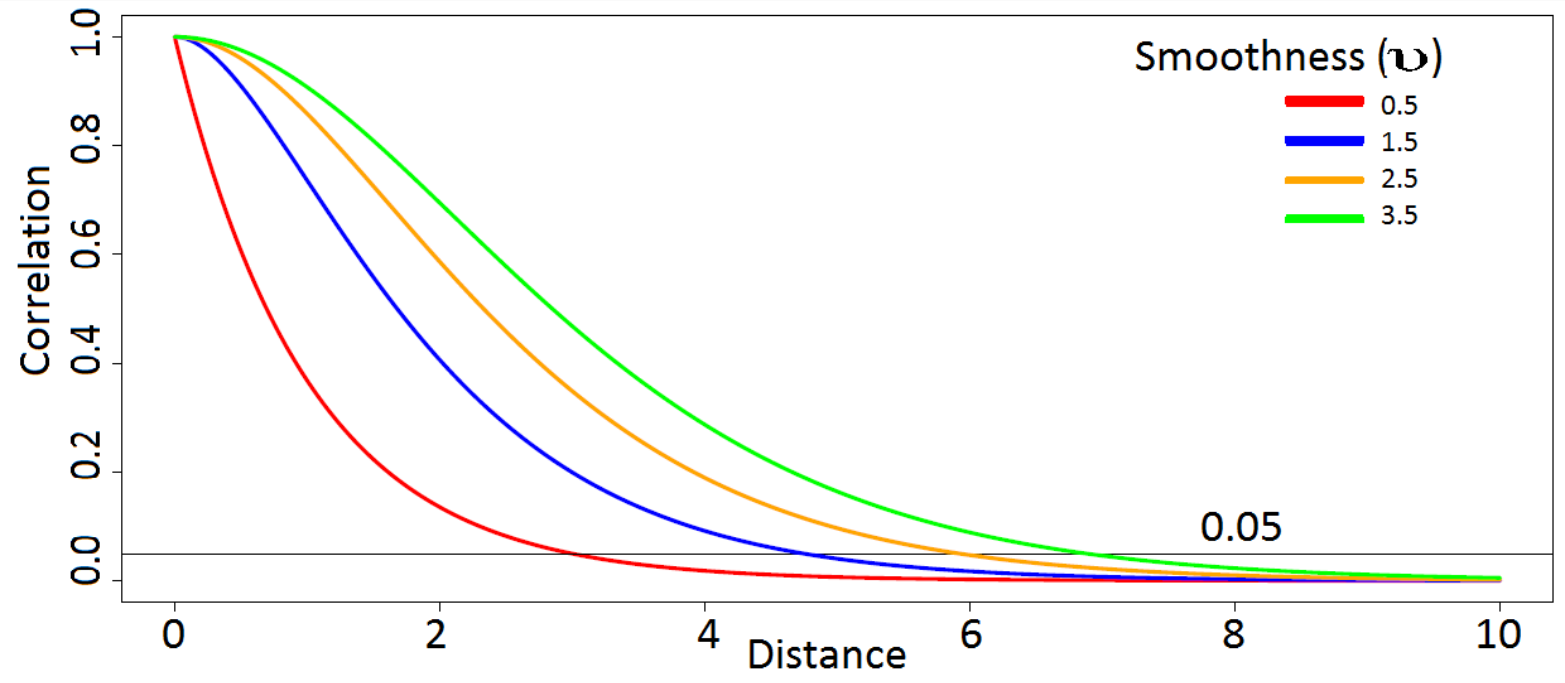
```
D[1,1] = K[1,1] and first row of A is zero.
for(i in 1:(n-1)) {
  Pa = N[i+1] # neighbors of i+1
  A[i+1,Pa] = solve(K[Pa,Pa], K[i+1,Pa])
  D[i+1,i+1] = K[i+1,i+1] - dot(K[i+1,Pa],A[i+1,Pa])
}
```

- We need to solve $n-1$ linear systems of size at most $m \times m$.
- We effectively model a (sparse) Cholesky factor instead of computing it.

Choosing neighbor sets

Matern Covariance Function:

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



Choosing neighbor sets

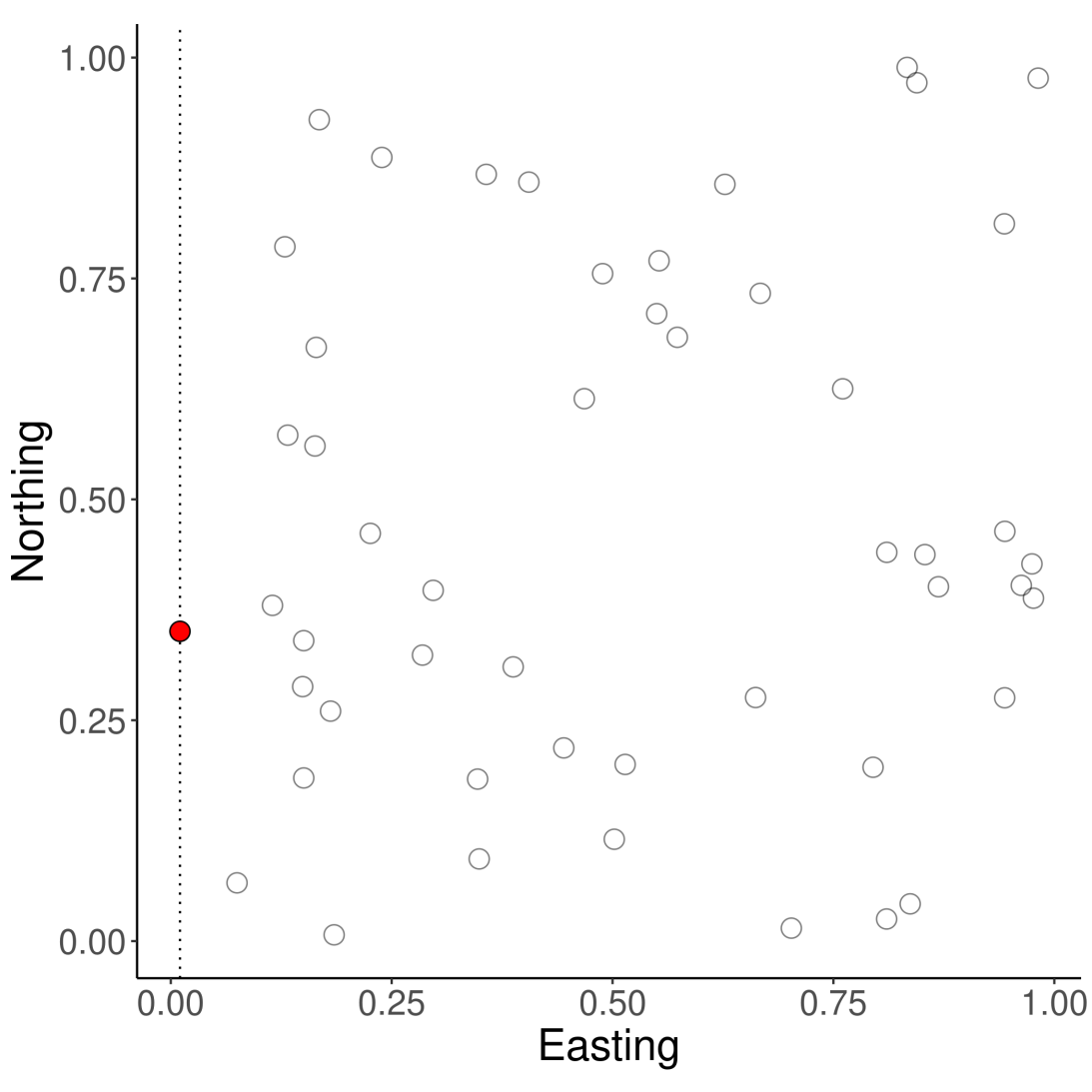
- Spatial covariance functions decay with distance

- Vecchia (1988): $N(\mathbf{s}_i) = m-$nearest neighbors of $\mathbf{s}_i$ in $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{i-1}$
  - Nearest points have highest correlations
  - Theory: "Screening effect" – Stein, 2002

- We use Vecchia's choice of m-nearest neighbor

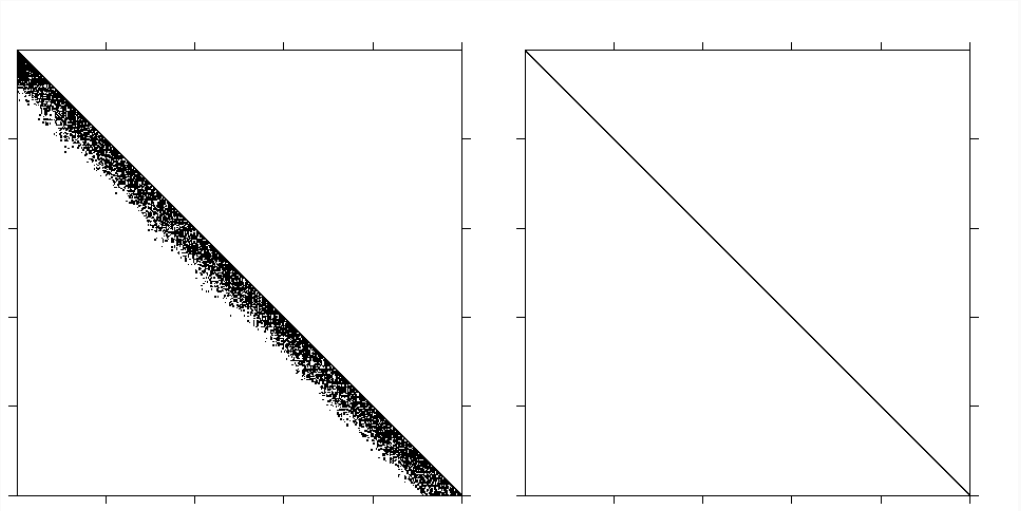
- Other choices proposed in Stein et al. (2004); Gramacy and Apley (2015); Guinness (2018) can also be used, with additional discussion in Finley et al. (2019) and Katzfuzz and Guinness (2021)

Nearest neighbors



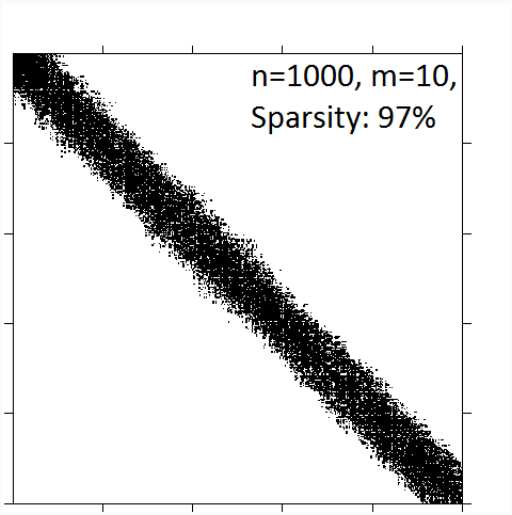
Sparse precision matrices

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



$\mathbf{I} - \mathbf{A}$

$\mathbf{D}^{-1}$



$\tilde{\mathbf{K}}_\theta^{-1}$

- $\det(\tilde{\mathbf{K}}_\theta) = \prod_{i=1}^{n} D_i$,
- $\tilde{\mathbf{K}}_\theta^{-1}$ is sparse with $O(nm^2)$ entries

Explore some $\mathbf{A}$ and $\tilde{\mathbf{K}}_\theta^{-1}$ sparsity patterns https://github.com/finleya/NNGP_LDL

Extension to a Process

- We have defined $\mathbf{w} \sim N(\mathbf{0}, \tilde{\boldsymbol{K}}_\theta)$ over the set of data locations $S = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$
- For $\mathbf{s} \notin S$, define $N(\mathbf{s})$ as set of m-nearest neighbors of $\mathbf{s}$ in S
- Define $w(\mathbf{s}) = \sum_{i:\mathbf{s}_i \in N(\mathbf{s})} a_i(\mathbf{s}) w(\mathbf{s}_i) + \eta(\mathbf{s})$ where $\eta(\mathbf{s}) \overset{ind}{\sim} N(0, d(\mathbf{s}))$
  - $a_i(\mathbf{s})$ and $d(\mathbf{s})$ 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

$$y(\mathbf{s}) = \mathbf{x}(\mathbf{s})^\top \boldsymbol{\beta} + w(\mathbf{s}) + \epsilon(\mathbf{s})$$

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

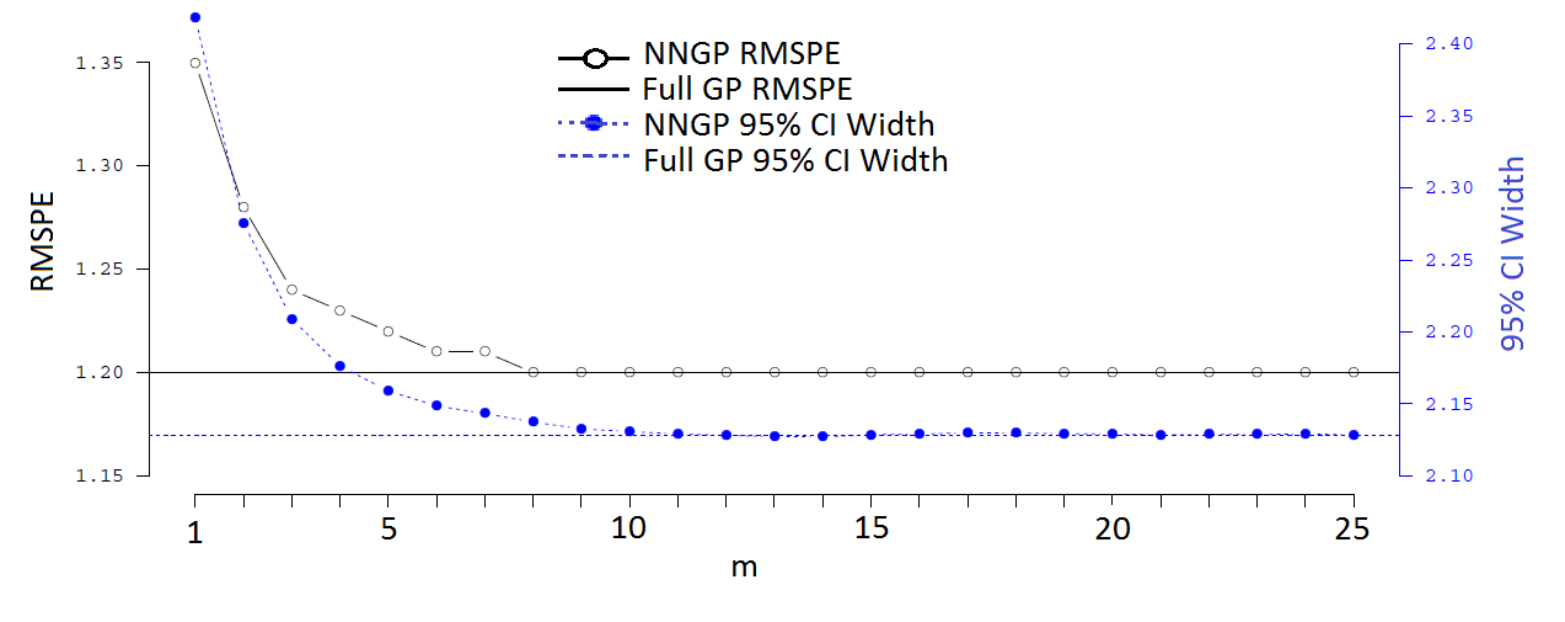
Full Bayesian Model

$$N(\mathbf{y} \mid \mathbf{X}\beta + \mathbf{w}, \tau^2\mathbf{I}) \times N(\mathbf{w} \mid \mathbf{0}, \tilde{\boldsymbol{K}}_\theta) \times N(\beta \mid \boldsymbol{\mu}_\beta, \mathbf{V}_\beta)$$
$$\times IG(\tau^2 \mid a_\tau, b_\tau) \times IG(\sigma^2 \mid a_\sigma, b_\sigma) \times Unif(\phi \mid a_\phi, b_\phi)$$

Gibbs sampler:

- Full conditionals for β, τ^2, σ^2 and $w(\mathbf{s}_i)$'s
- Metropolis step for updating ϕ
- Posterior predictive distribution at any location using composition sampling

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

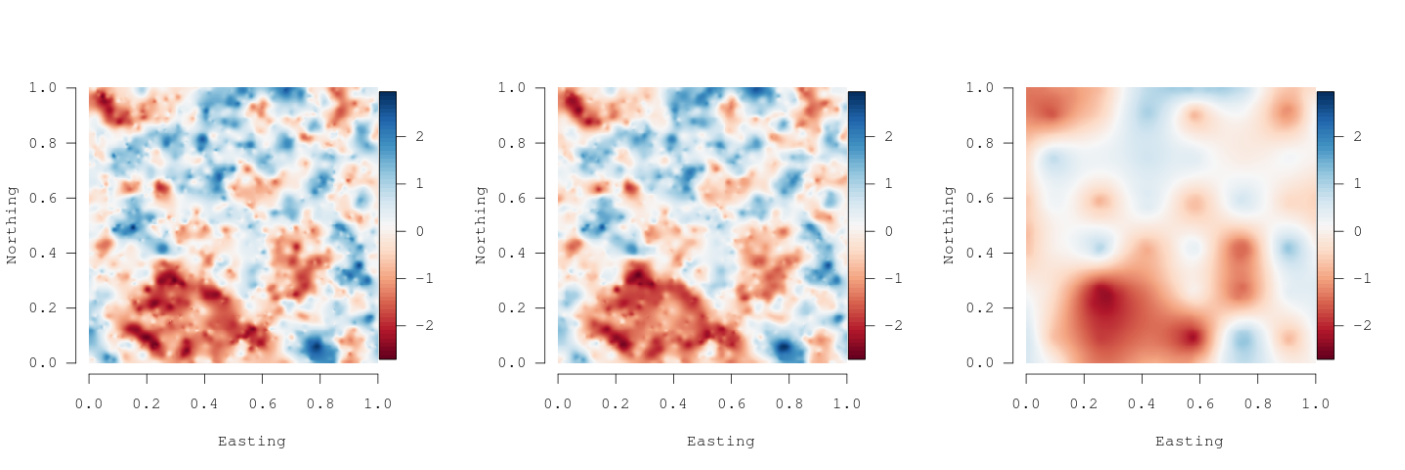
Storage and computation

- 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

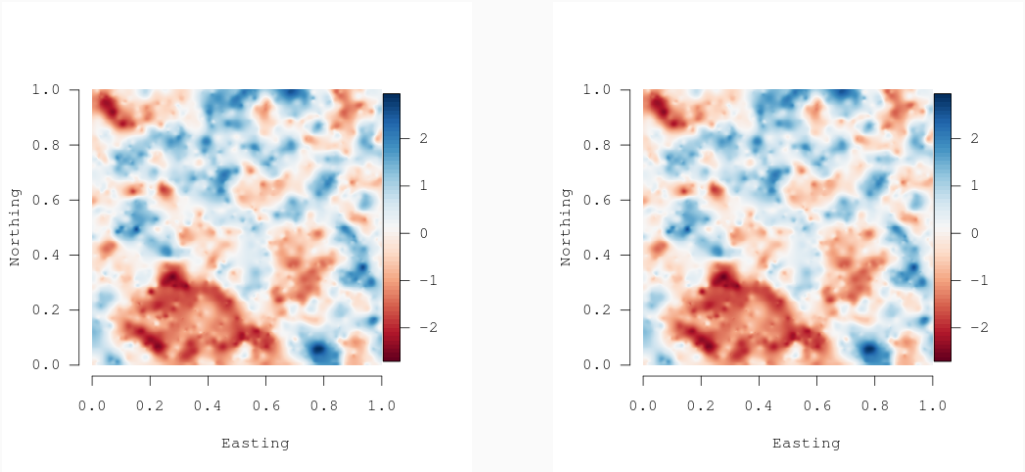
Simulation experiment

- 2500 locations on a unit square
- $y(\mathbf{s}) = \beta_0 + \beta_1 x(\mathbf{s}) + w(\mathbf{s}) + \epsilon(\mathbf{s})$
- Single covariate $x(\mathbf{s})$ generated as iid $N(0, 1)$
- Spatial effects generated from $GP(0, \sigma^2 R(\cdot, \cdot \mid \phi))$
- $R(\cdot, \cdot \mid \phi)$ 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

Multvariate spatial linear model

- Spatial linear model for q-variate spatial data:
  $y_i(\mathbf{s}) = \mathbf{x}_i^\top(\mathbf{s})\beta_i + w_i(s) + \epsilon_i(s)$ for $i = 1, 2, \ldots, q$
- $\boldsymbol{\epsilon}(s) = (\epsilon_1(s), \epsilon_2(s), \ldots, \epsilon_q(s))^\top \sim N(0, E)$ where E is the $q \times q$ noise matrix
- $\mathbf{w}(s) = (w_1(s), w_2(s), \ldots, w_q(s))^\top$ is modeled as a q-variate Gaussian process

Multivariate GPs

- $Cov(w(\mathbf{s}_i), w(\mathbf{s}_j)) = K(s_i, s_j \mid \boldsymbol{\theta})$ – a $q \times q$ cross-covariance matrix
- Choices for the function $K(\cdot, \cdot \mid \boldsymbol{\theta})$
  - Multivariate Matérn
  - Linear model of co-regionalization
- For data observed at n locations, all choices lead to a dense $nq \times nq$ matrix $\mathbf{K}_\theta = Cov(w(\mathbf{s}_1), w(\mathbf{s}_2), \ldots, w(\mathbf{s}_n))$
- Not scalable when nq is large

Multivariate NNGPs

- Cholesky factor approach similar to the univariate case

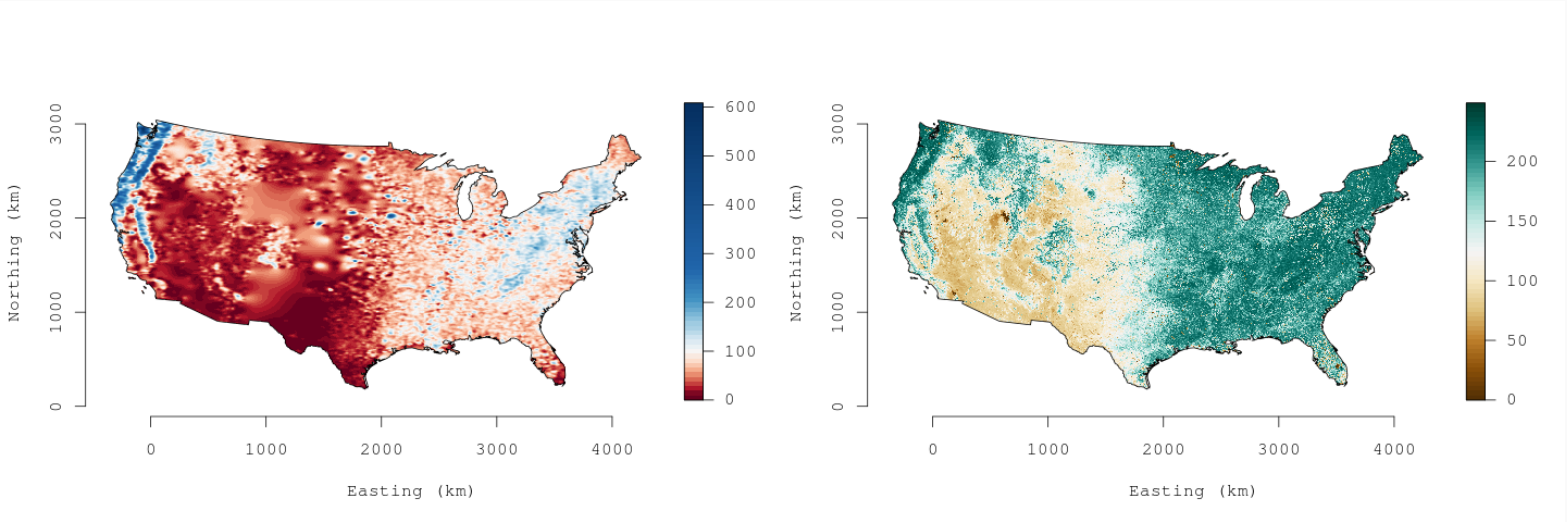
$$\begin{bmatrix} \mathbf{w}(\mathbf{s}_1) \\ \mathbf{w}(\mathbf{s}_2) \\ \mathbf{w}(\mathbf{s}_3) \\ \vdots \\ \mathbf{w}(\mathbf{s}_n) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ \mathbf{A}_{21} & 0 & 0 & \dots & 0 & 0 \\ \mathbf{A}_{31} & \mathbf{A}_{32} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{A}_{n1} & \mathbf{A}_{n2} & \mathbf{A}_{n3} & \dots & \mathbf{A}_{n,n-1} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}(\mathbf{s}_1) \\ \mathbf{w}(\mathbf{s}_2) \\ \mathbf{w}(\mathbf{s}_3) \\ \vdots \\ \mathbf{w}(\mathbf{s}_n) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\eta}(\mathbf{s}_1) \\ \boldsymbol{\eta}(\mathbf{s}_2) \\ \boldsymbol{\eta}(\mathbf{s}_3) \\ \vdots \\ \boldsymbol{\eta}(\mathbf{s}_n) \end{bmatrix}$$

$$\implies \mathbf{w} = \mathbf{A}\mathbf{w} + \boldsymbol{\eta}; \quad \boldsymbol{\eta} \sim N(\mathbf{0}, \mathbf{D}),\ \mathbf{D} = diag(\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_n).$$

Only differences:

- $\mathbf{w}(\mathbf{s}_i)$ and $\boldsymbol{\eta}(\mathbf{s}_i)$'s are $q \times 1$ vectors and $\mathbf{A}_{ij}$ and $\mathbf{D}_i$'s are $q \times q$ matrix
- we must solve $n-1$ at most $mq \times mq$ linear systems (challenging when q gets large, e.g., $q > 5$).

U.S. Forest biomass data



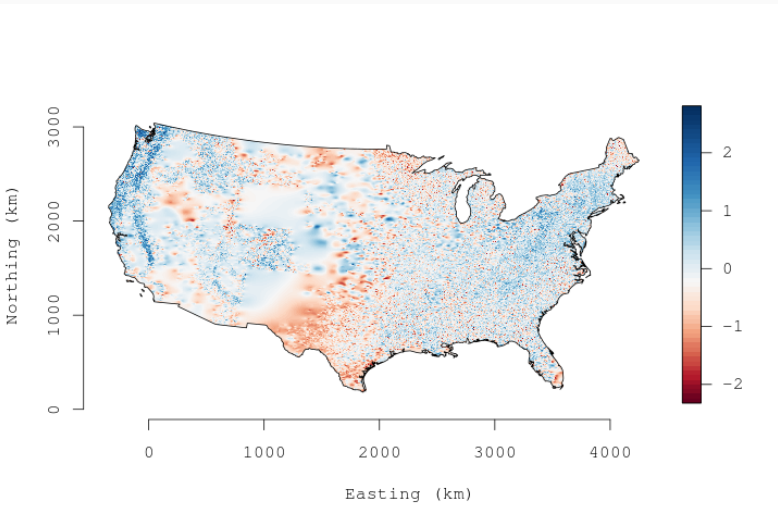
Observed biomass NDVI

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

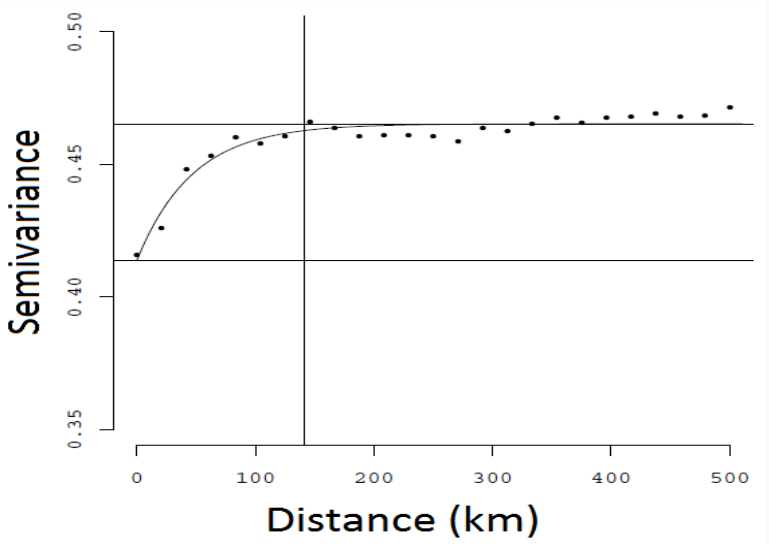
U.S. Forest biomass data

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

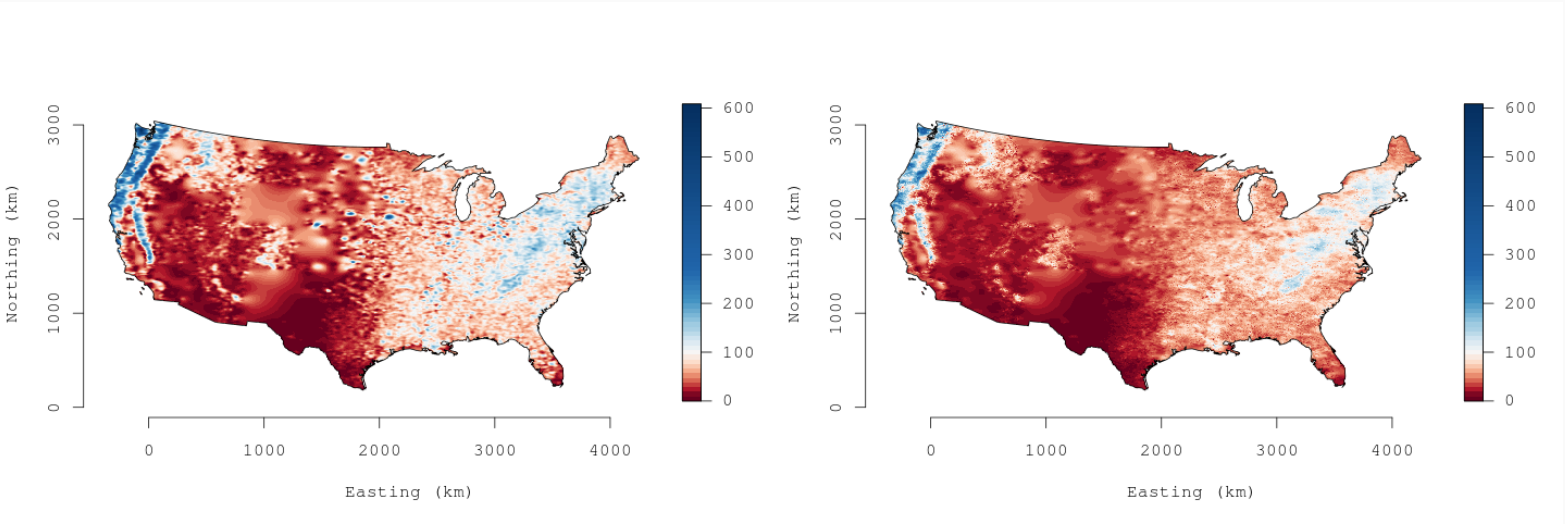
Forest biomass dataset

- $n \approx 10^5$ (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

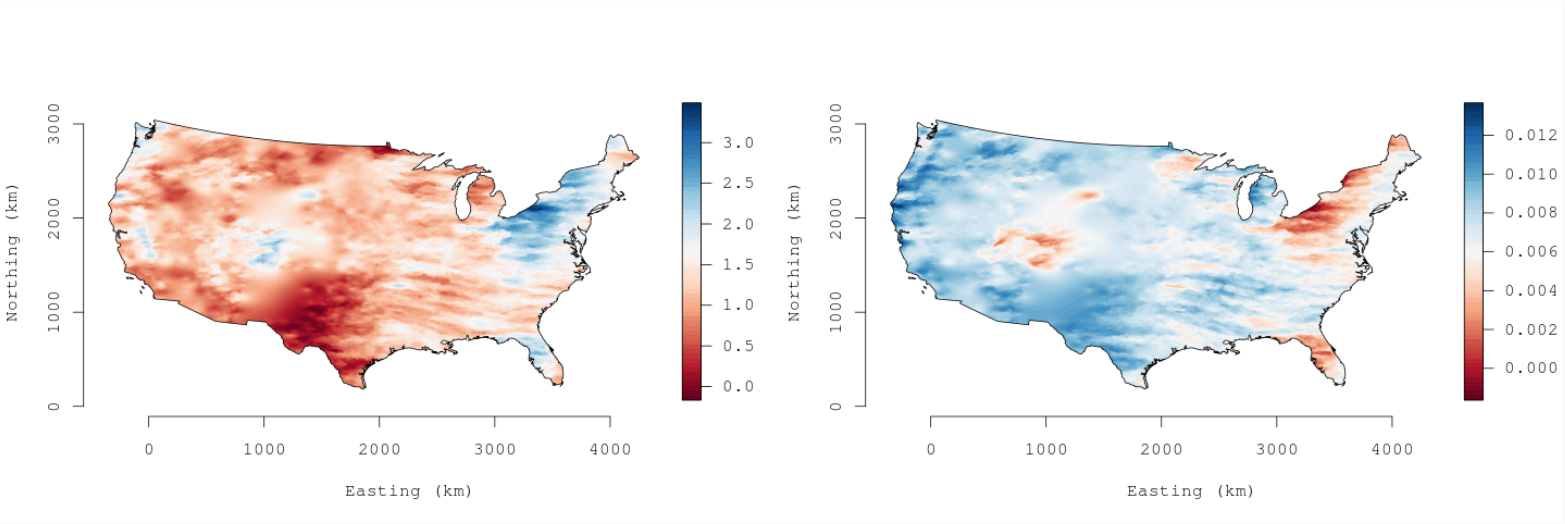
- $Biomass(s) = \beta_0(s) + \beta_1(s) NDVI(s) + \epsilon(s)$
- $\mathbf{w}(s) = (\beta_0(s), \beta_1(s))^\top \sim$ Bivariate $NNGP(0, \tilde{K}(\cdot, \cdot \mid \theta))$, $m = 5$
- Time $\approx$ 6 seconds per iteration
- Full inferential output: 41 hours (25000 MCMC iterations)

Forest biomass data



Observed biomass

Fitted biomass



$\beta_0(s)$

$\beta_{NDVI}(s)$

Reducing parameter dimensionality

- The Gibbs sampler algorithm for the NNGP updates $w(\mathbf{s}_1), w(\mathbf{s}_2), \ldots, w(\mathbf{s}_n)$ 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:

$$
\begin{aligned}
y(\mathbf{s}) &= \mathbf{x}(\mathbf{s})^\top \boldsymbol{\beta} + w(\mathbf{s}) + \epsilon(\mathbf{s}) \\
w(\mathbf{s}) &\sim NNGP(0, K(\cdot,\cdot \mid \boldsymbol{\theta})) \\
\epsilon(\mathbf{s}) &\overset{\text{iid}}{\sim} N(0, \tau^2)
\end{aligned}
$$

- Latent model: $\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta} + \mathbf{w}, \tau^2\mathbf{I}); \mathbf{w} \sim N(\mathbf{0}, \tilde{\boldsymbol{K}}_\theta)$
- Collapsed model: Marginalizing out $\mathbf{w}$, $\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \tau^2\mathbf{I} + \tilde{\boldsymbol{K}}_\theta)$

Collapsed NNGP

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \tau^2\mathbf{I} + \tilde{\boldsymbol{K}}_\theta)$$

- Only involves few parameters $\boldsymbol{\beta}$, τ^2 and $\boldsymbol{\theta} = (\sigma^2, \phi)$
- Drastically reduces the MCMC dimensionality
- Gibbs sampler updates are based on sparse linear systems using $\tilde{\boldsymbol{K}}_\theta^{-1}$ (e.g., use `CHOLMOD`)
- Improved MCMC convergence
- Can recover posterior distribution of $\mathbf{w} \,|\, \mathbf{y}$
- Complexity of the algorithm depends on the design of the data locations and is not guaranteed to be $O(n)$

Response NNGP

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

Conjugate NNGP

- Full GP model: $\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = \sigma^2\mathbf{M}$
- $\mathbf{M} = \mathbf{R}(\phi) + \alpha\mathbf{I}$
- $\alpha = \tau^2/\sigma^2$ is the ratio of the noise to signal variance
- $\tilde{\boldsymbol{\Sigma}} = \sigma^2\tilde{\mathbf{M}}$ where $\tilde{\mathbf{M}}$ is the NNGP approximation for $\mathbf{M}$

Conjugate NNGP

- Full GP model: $\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = \sigma^2\mathbf{M}$
- $\mathbf{M} = \mathbf{R}(\phi) + \alpha\mathbf{I}$
- $\alpha = \tau^2/\sigma^2$ is the ratio of the noise to signal variance
- $\tilde{\boldsymbol{\Sigma}} = \sigma^2\tilde{\mathbf{M}}$ where $\tilde{\mathbf{M}}$ is the NNGP approximation for $\mathbf{M}$

- If ϕ and α are known, $\mathbf{M}$, and hence $\tilde{\mathbf{M}}$, are known matrices
- The model becomes a standard Bayesian linear model
- Assume a *Normal Inverse Gamma* prior for $(\boldsymbol{\beta}, \sigma^2)^\top$
- $(\boldsymbol{\beta}, \sigma^2)^\top \sim NIG(\boldsymbol{\mu}_\beta, \mathbf{V}_\beta, a_\sigma, b_\sigma)$, i.e., $\boldsymbol{\beta} \,|\, \sigma^2 \sim N(\boldsymbol{\mu}_\beta, \sigma^2\mathbf{V}_\beta)$ and $\sigma^2 \sim IG(a_\sigma, b_\sigma)$
- Exact posterior distributions of β and σ^2 are available

Can handle n in the 100s of millions!

Comparison of NNGP models

	Latent	Collapsed	Response	Conjugate
$O(n)$ time	Yes	No	Yes	Yes
Recovery of $\mathbf{w} \mid \mathbf{y}$	Yes	Yes	No	Yes
Parameter dimensionality	High	Low	Low	Low
Inference on θ	Yes	Yes	Yes	Partially

Comparison of NNGP models

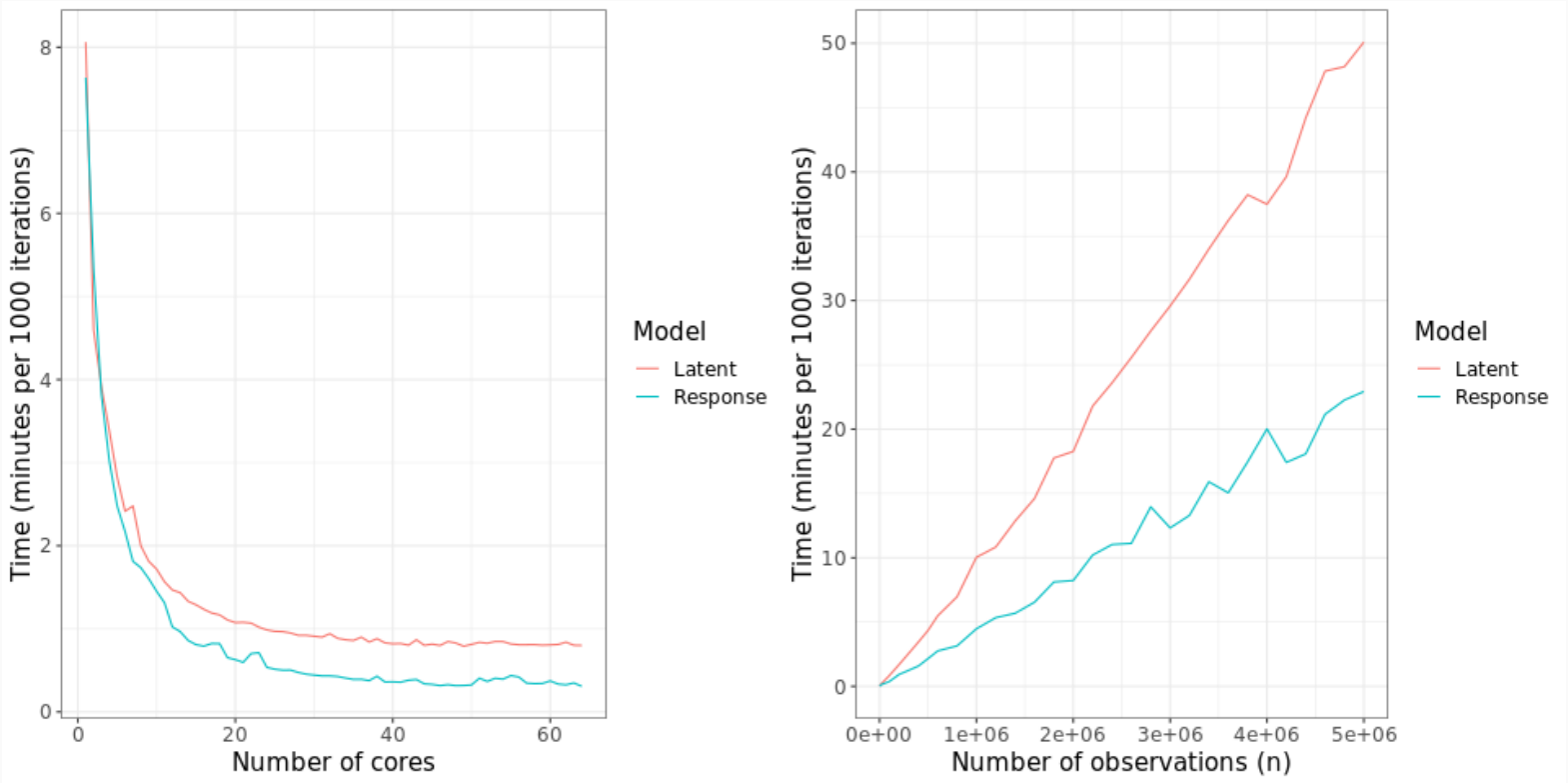


Figure: (a) Runtime for 1000 MCMC iterations for $n = 100000$ and different number of cores. (b) Runtime for 1000 MCMC iterations using 40 cores and n from 1000 to 5 million. Model type (latent and response) refers to different NNGP parameterizations, see Finley et al. 2022.

Summary of Nearest Neighbor Gaussian Processes

- Sparsity inducing Gaussian process
- Constructed from sparse Cholesky factors based on m nearest neighbors
- Scalability in 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
- R packages `spNNGP` (Finley et al. 2022) and `spOccupancy` (Doser et al., 2022) on CRAN