Scalable methods for large spatial data: Nearest Neighbor Gaussian processes

Abhi Datta

Department of Biostatistics, Bloomberg School of Public Health, Johns Hopkins University, Baltimore, Maryland

Review of Low rank Gaussian Predictive Process

Pros

- Choose knots $S^* = \{s_1^*, s_2^*, \ldots, s_r^*\}$
- $\tilde{w}(s) = E(w(s) \mid w(S^*)) + \eta(s)$
- $var(\tilde{w}) = A_{n\times r} Var(\tilde{w}(S^*))_{r\times r} A' + D_{n\times n}$
- Matrix identities ensure we only need to invert the $r \times r$ matrix
- Computationally tractable – FLOPs count $O(nr^2)$
- 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

Review of Low rank Gaussian Predictive Process

Cons

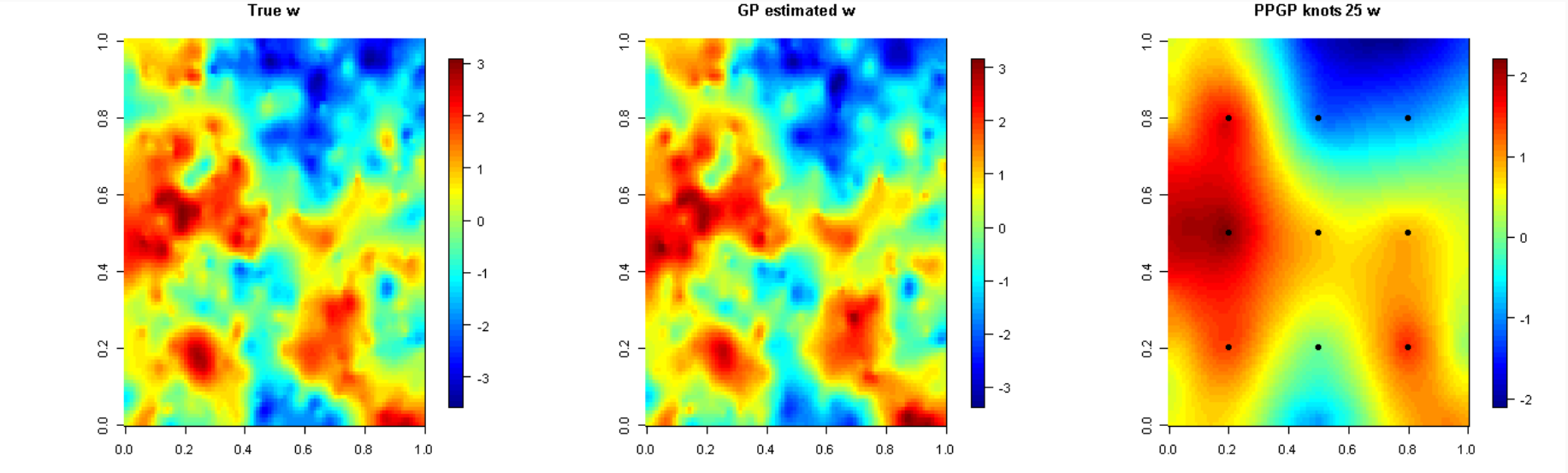


Figure: Comparing full GP vs low-rank GP

- 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:

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

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

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

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 $\Rightarrow$

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

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

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 $\Rightarrow$

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

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

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

Cholesky factors

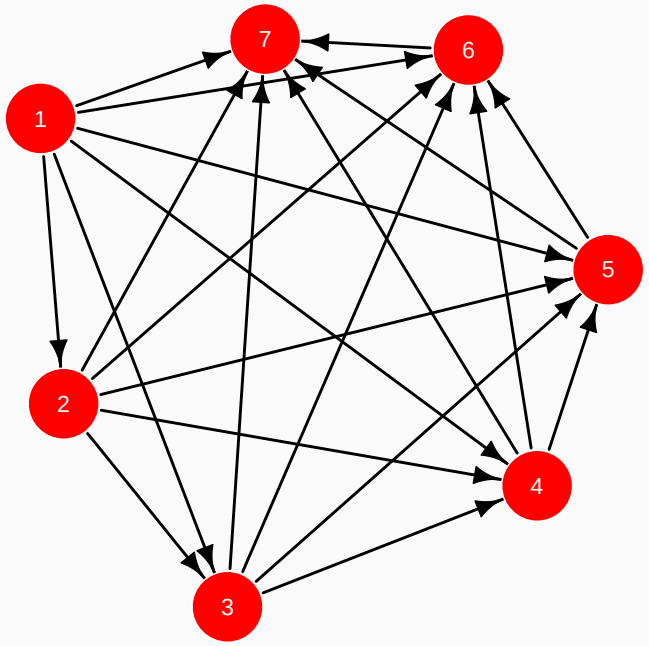
- $w_{<i} = (w_1, w_2, \ldots, w_{i-1})'$
- $c_i = Cov(w_i, w_{<i})$, $C_i = Var(w_{<i})$
- i^{th} row of A and $d_i = Var(\eta_i)$ are obtained from $p(w_i \mid w_{<i})$ as follows:
  - Solve for a_{ij}'s from $\sum_{j=1}^{i-1} a_{ij} w_j = E(w_i \mid w_{<i}) = c_i' C_i^{-1} w_{<i}$

- 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

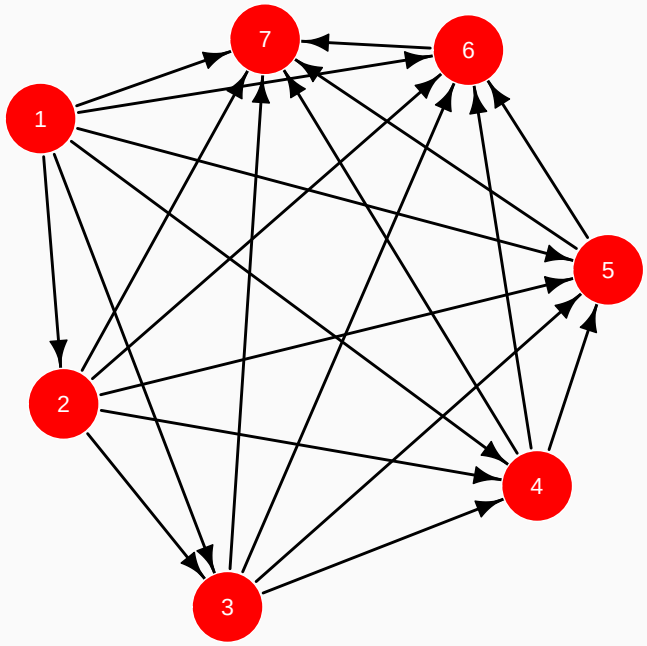
- $w_{<i} = (w_1, w_2, \ldots, w_{i-1})'$
- $c_i = Cov(w_i, w_{<i})$, $C_i = Var(w_{<i})$
- i^{th} row of A and $d_i = Var(\eta_i)$ are obtained from $p(w_i \mid w_{<i})$ as follows:
  - Solve for a_{ij}'s from $\sum_{j=1}^{i-1} a_{ij} w_j = E(w_i \mid w_{<i}) = c_i' C_i^{-1} w_{<i}$
  - $d_i = Var(w_i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



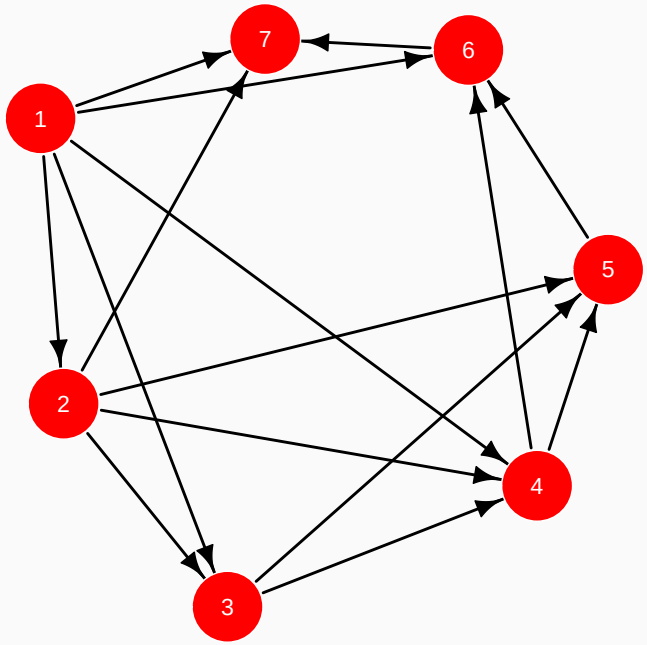
- Number of non-zero entries (sparsity) of A equals number of arrows in the graph
- In particular: Sparsity of the i^{th} row of A is same as the number of arrows towards i in the DAG

Introducing sparsity via graphical models



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

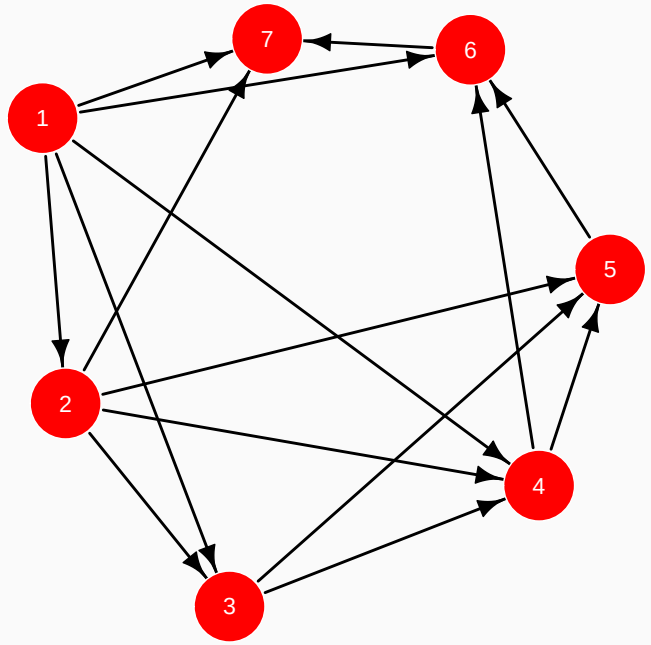
Introducing sparsity via graphical models



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

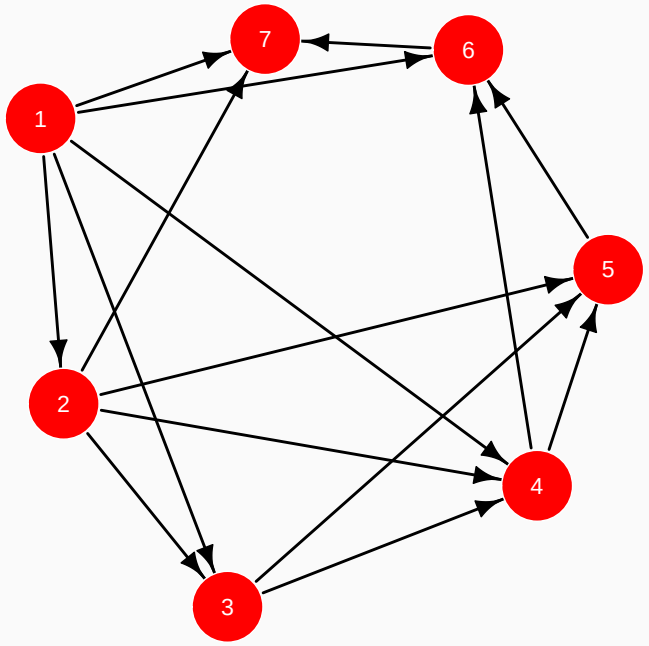
$$p(y_5 \mid \cancel{y_1}, y_2, y_3, y_4)p(y_6 \mid y_1, \cancel{y_2}, \cancel{y_3}, y_4, y_5)p(y_7 \mid y_1, y_2, \cancel{y_3}, \cancel{y_4}, \cancel{y_5}, y_6)$$

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:

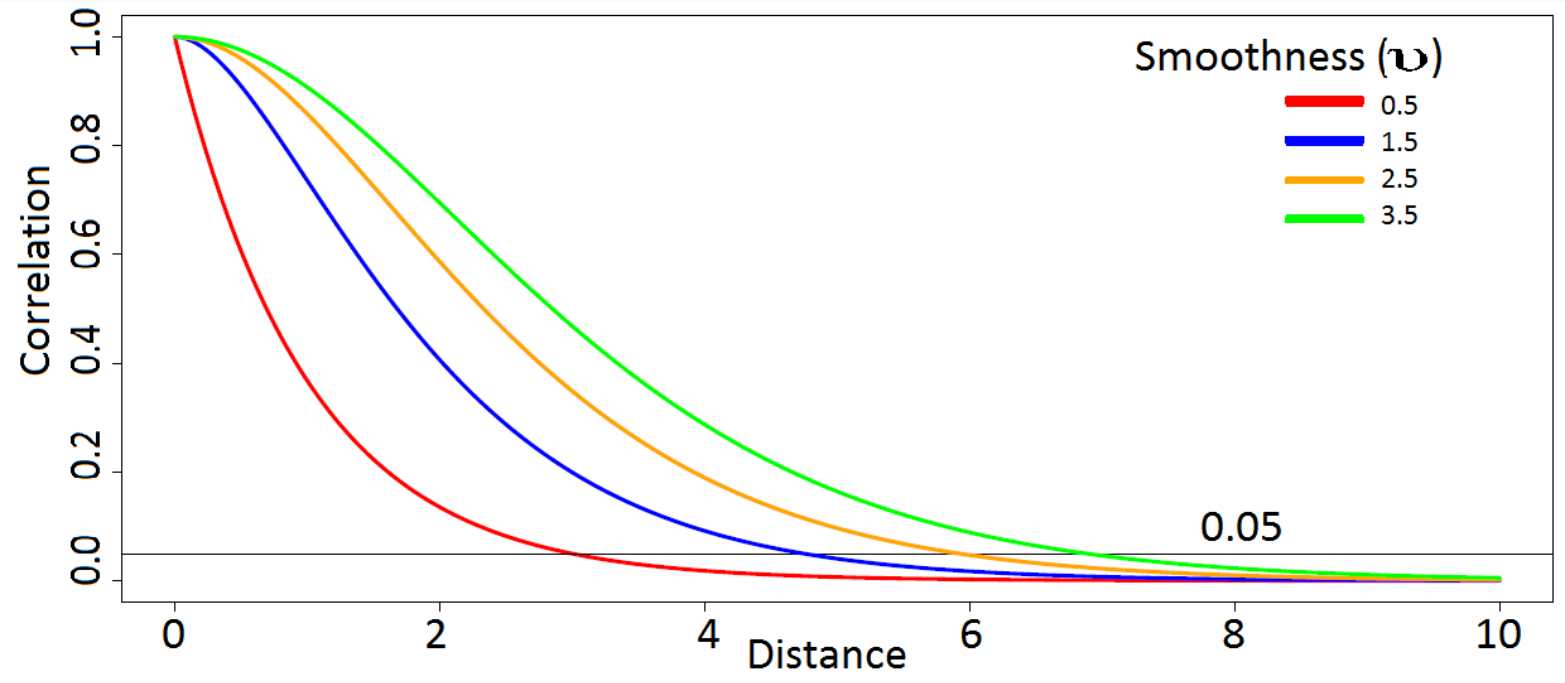
$$\mathrm{E}[w_i \mid w_{N(i)}] = \sum_{j \in N(i)} a_{ij} w_j$$

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

Choosing neighbor sets

Matern Covariance Function:

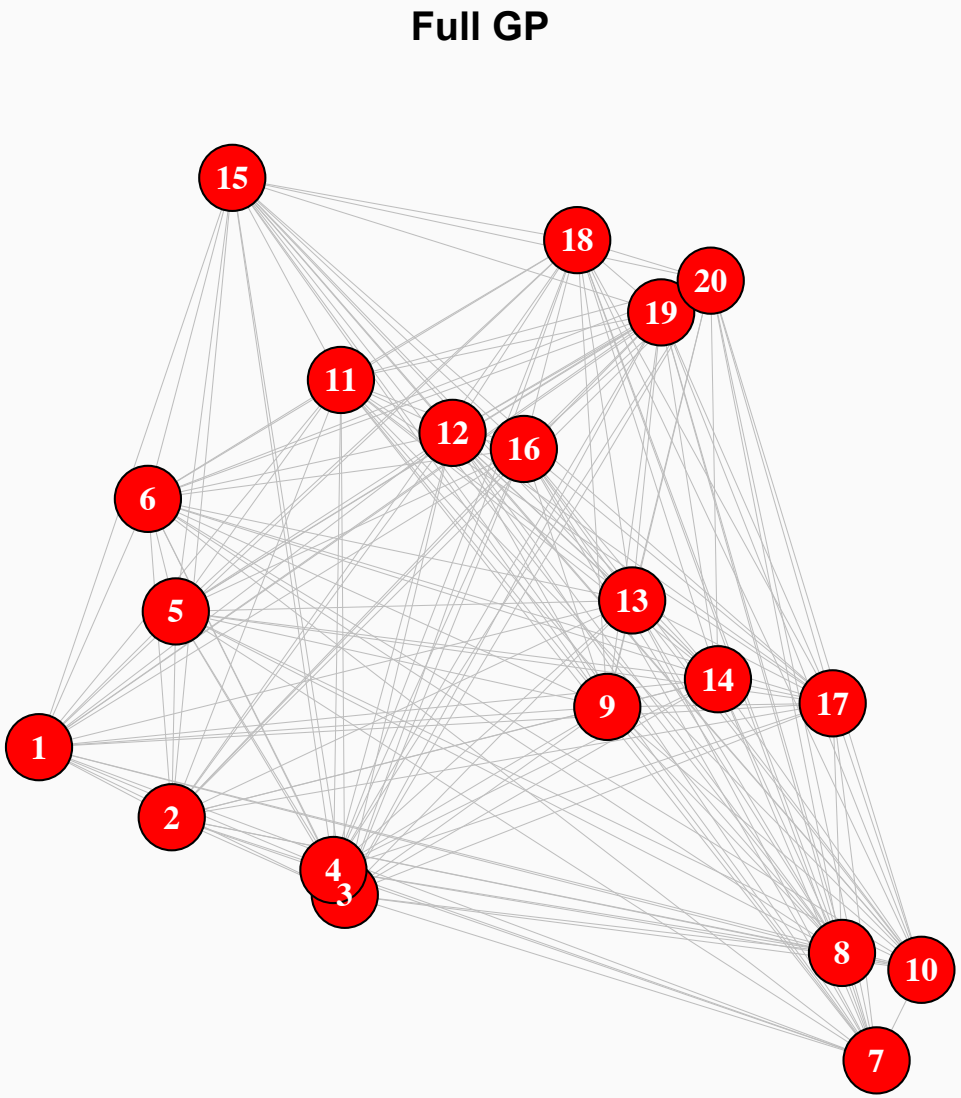
$$C(s_i, s_j) = \frac{1}{2^{\nu-1}\Gamma(\nu)}(||s_i - s_j||\phi)^{\nu}\mathscr{K}_{\nu}(||s_i - s_j||\phi);\ \phi > 0, \nu > 0,$$



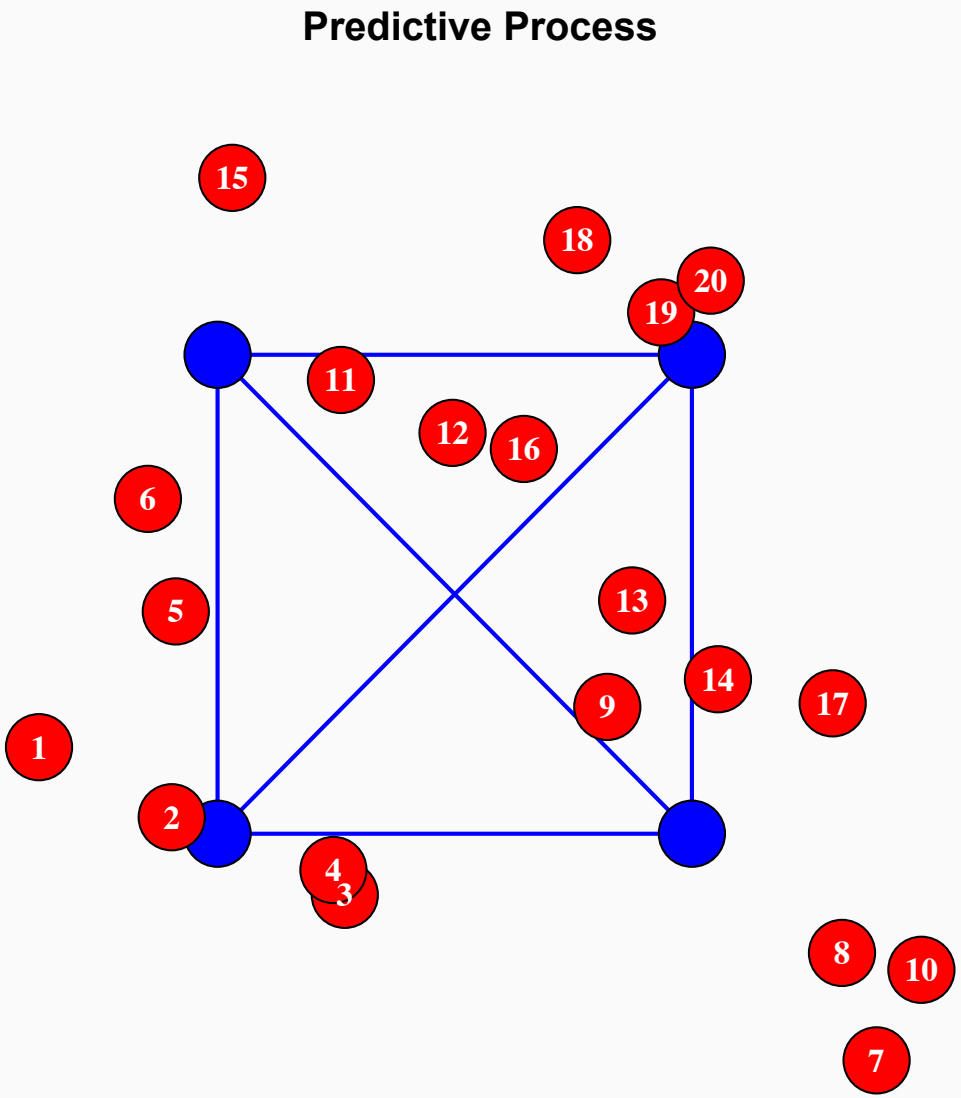
Choosing neighbor sets

- 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

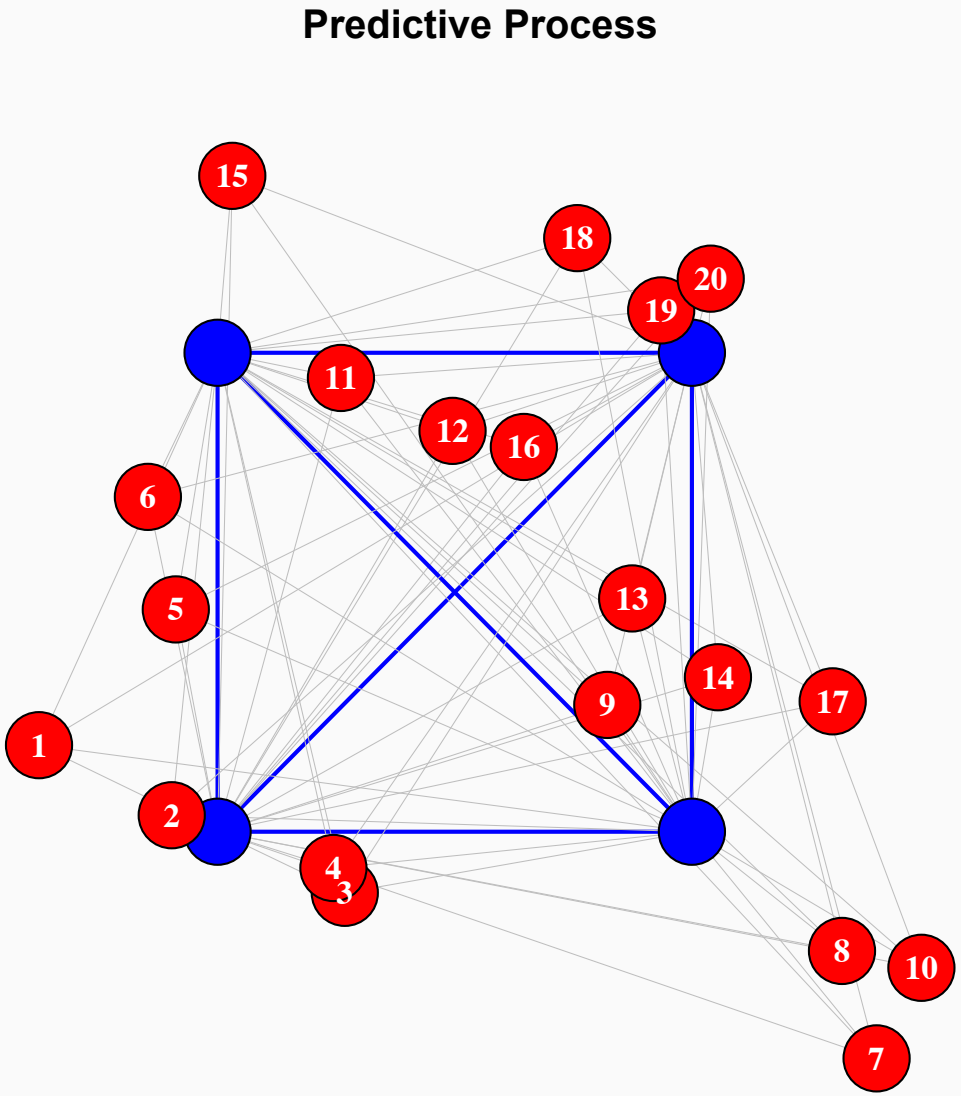
Graohical models: Full GP vs low rank vs NNGP



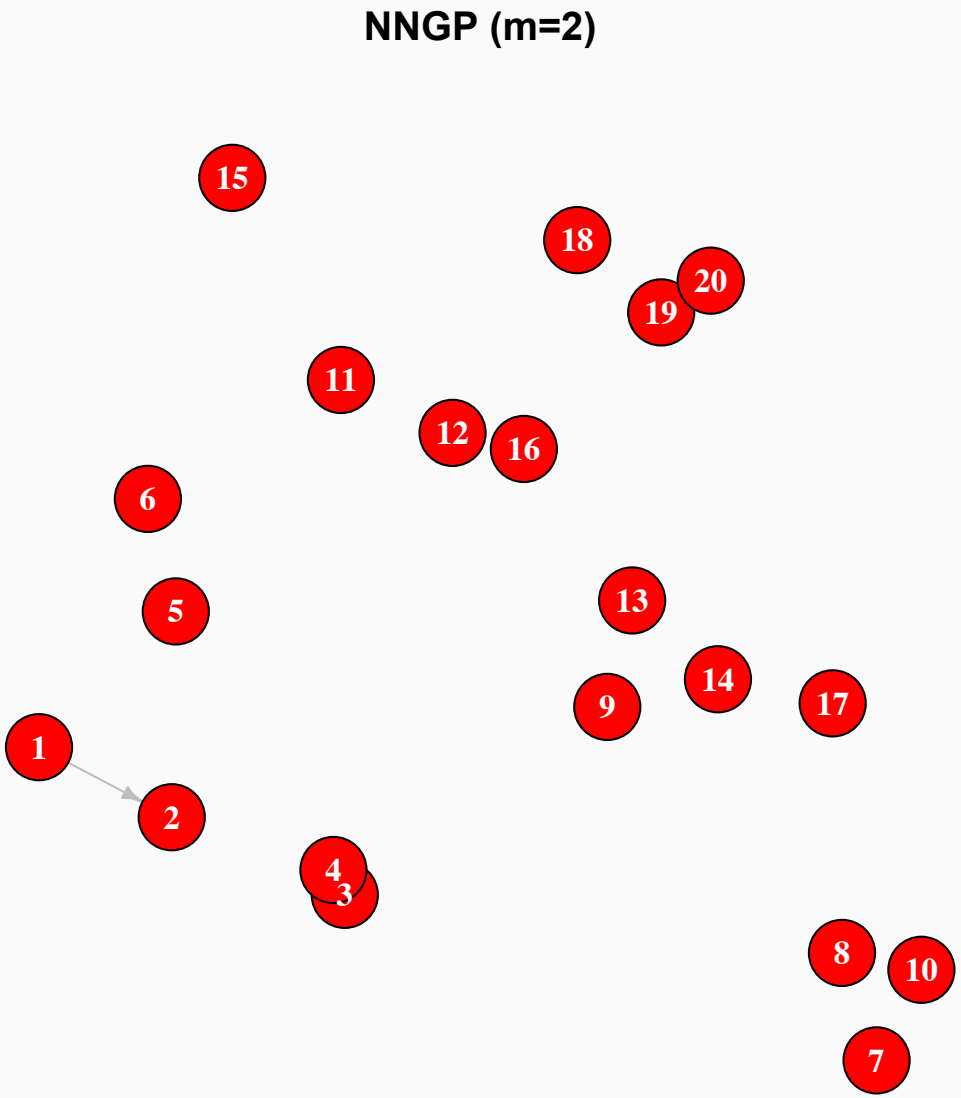
Graohical models: Full GP vs low rank vs NNGP



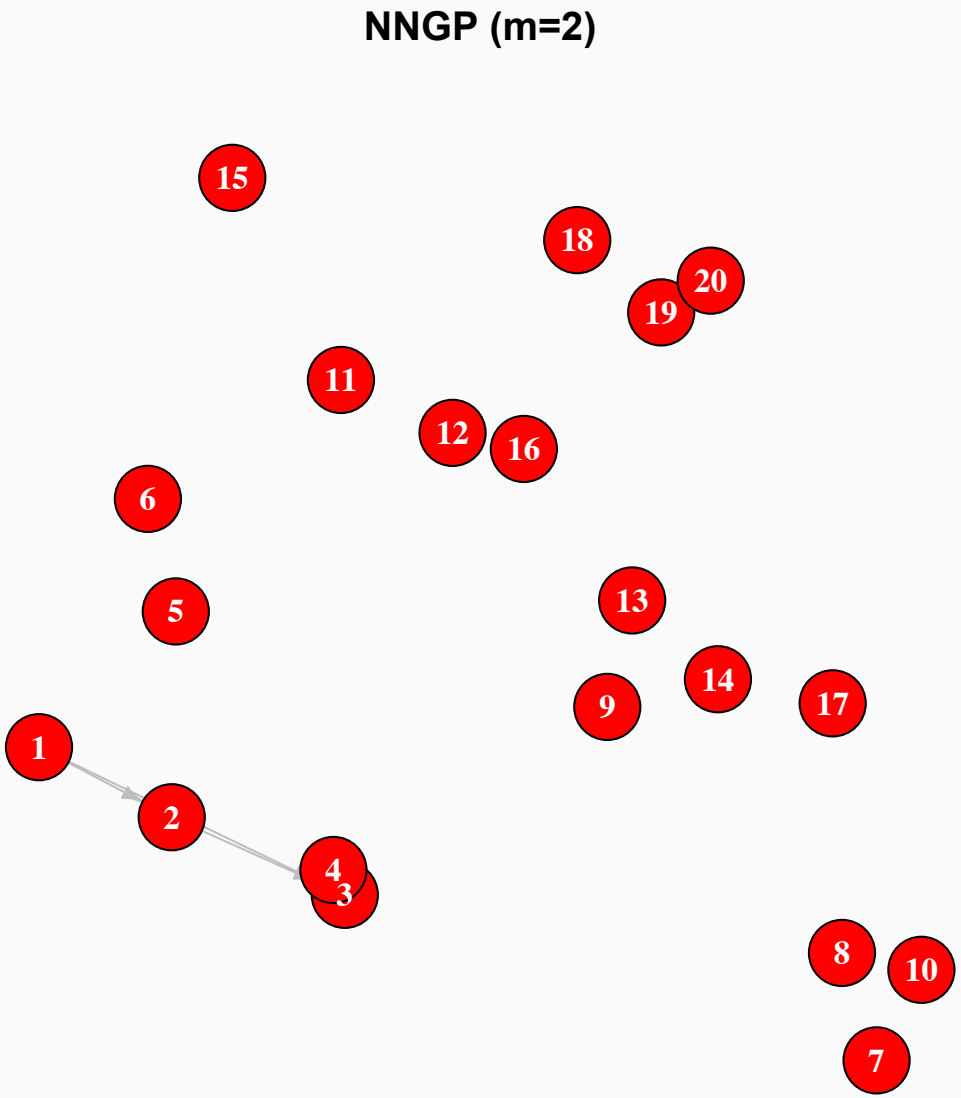
Graohical models: Full GP vs low rank vs NNGP



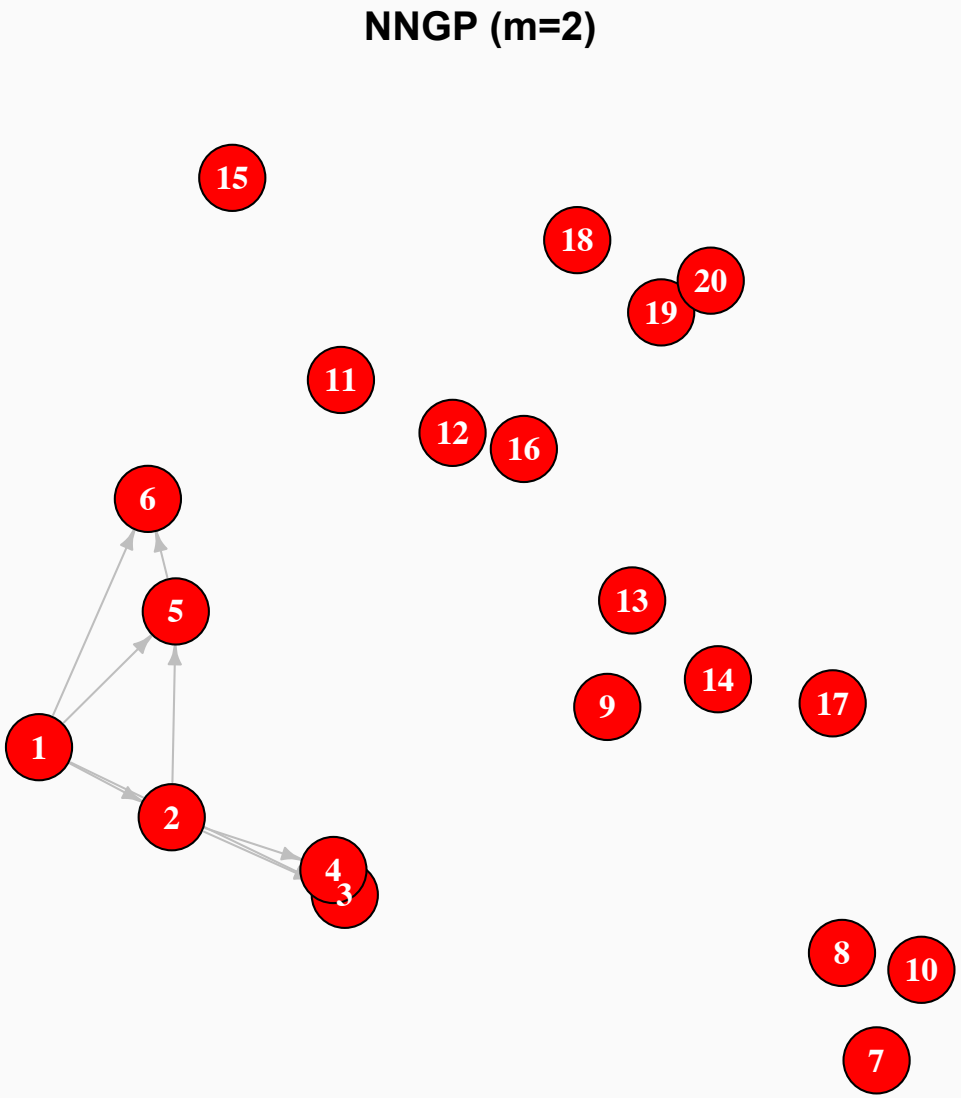
Graohical models: Full GP vs low rank vs NNGP



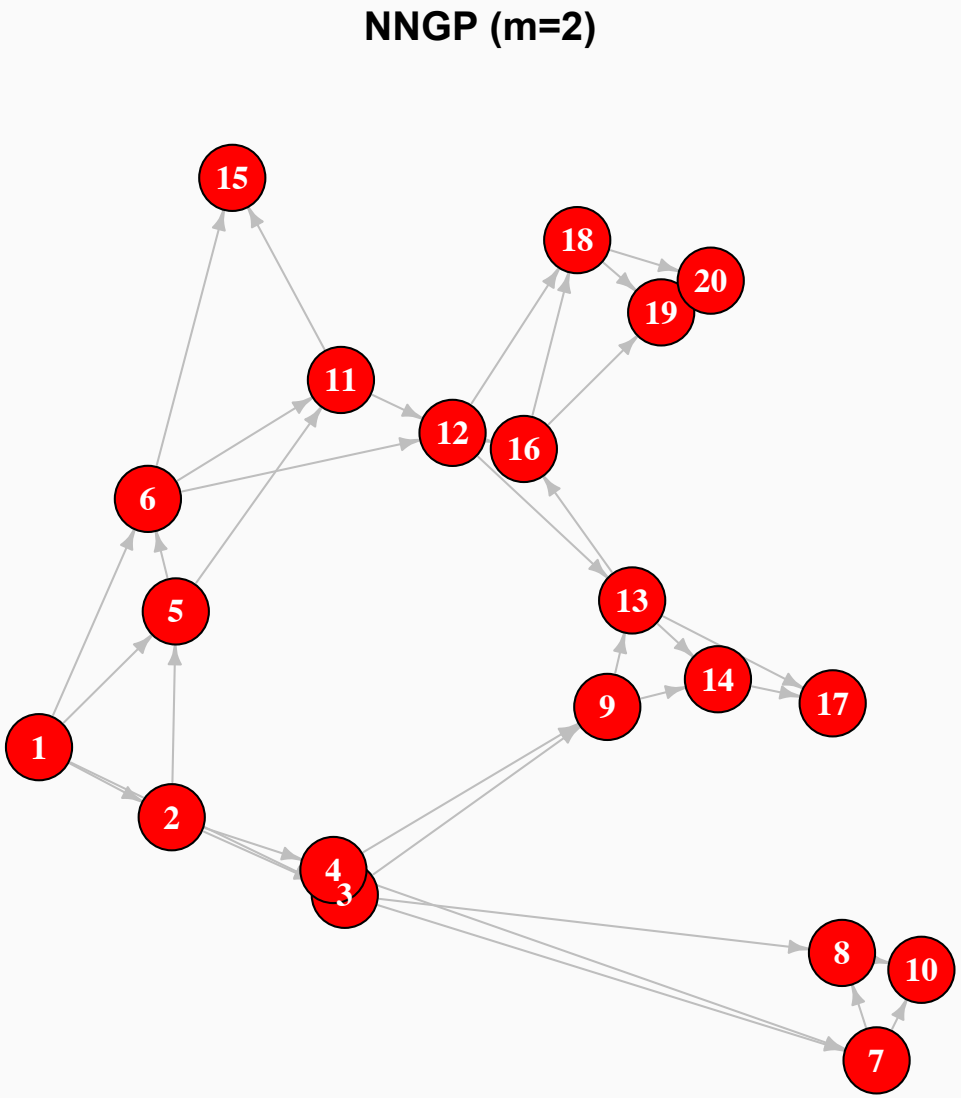
Graohical models: Full GP vs low rank vs NNGP



Graohical models: Full GP vs low rank vs NNGP

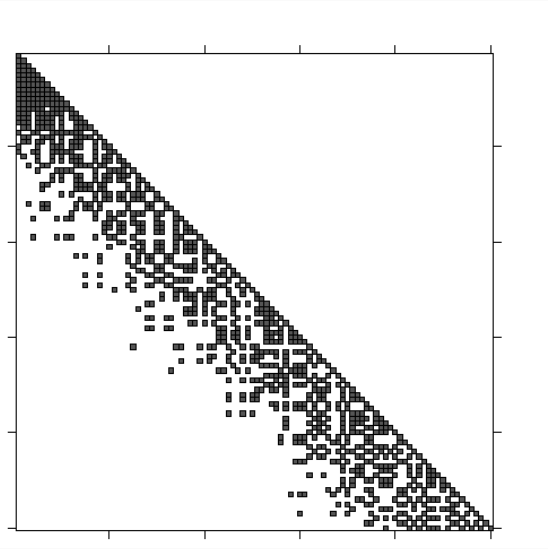


Graohical models: Full GP vs low rank vs NNGP

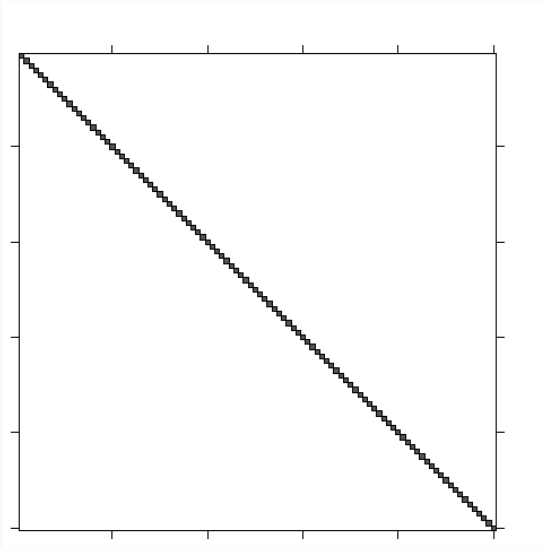


Sparse precision matrices

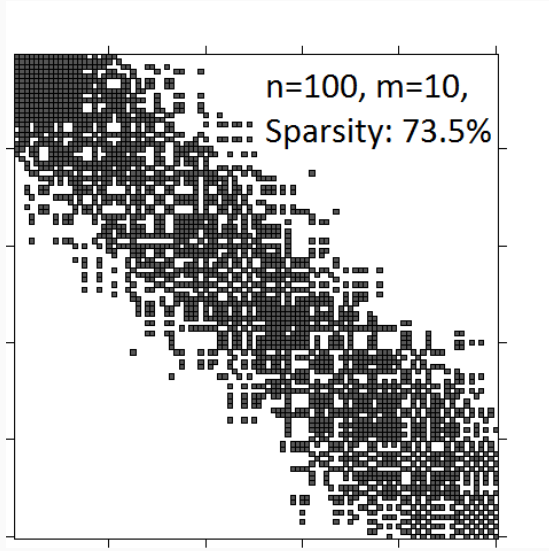
- The neighbor sets and the covariance function $C(\cdot, \cdot)$ define a sparse Cholesky factor A
- $N(w \mid 0, C) \approx N(w \mid 0, \tilde{C})$; $\tilde{C}^{-1} = (I - A)^\top D^{-1}(I - A)$



A



D

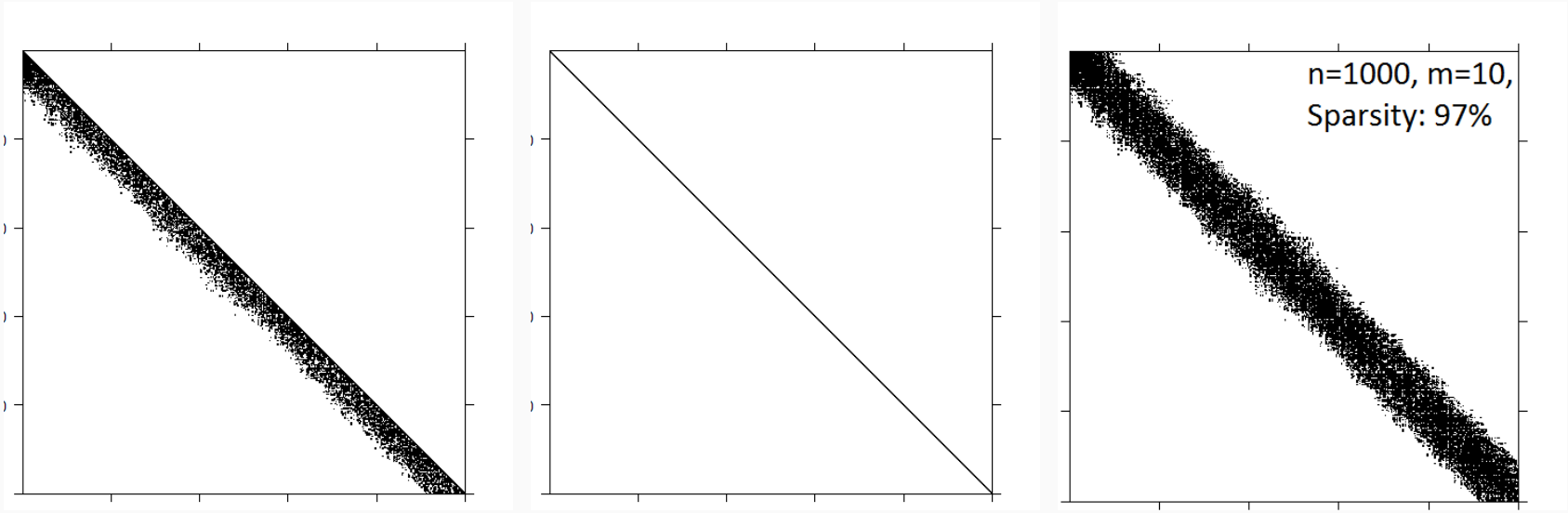


$\tilde{C}^{-1}$

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

Sparse precision matrices

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



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

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

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

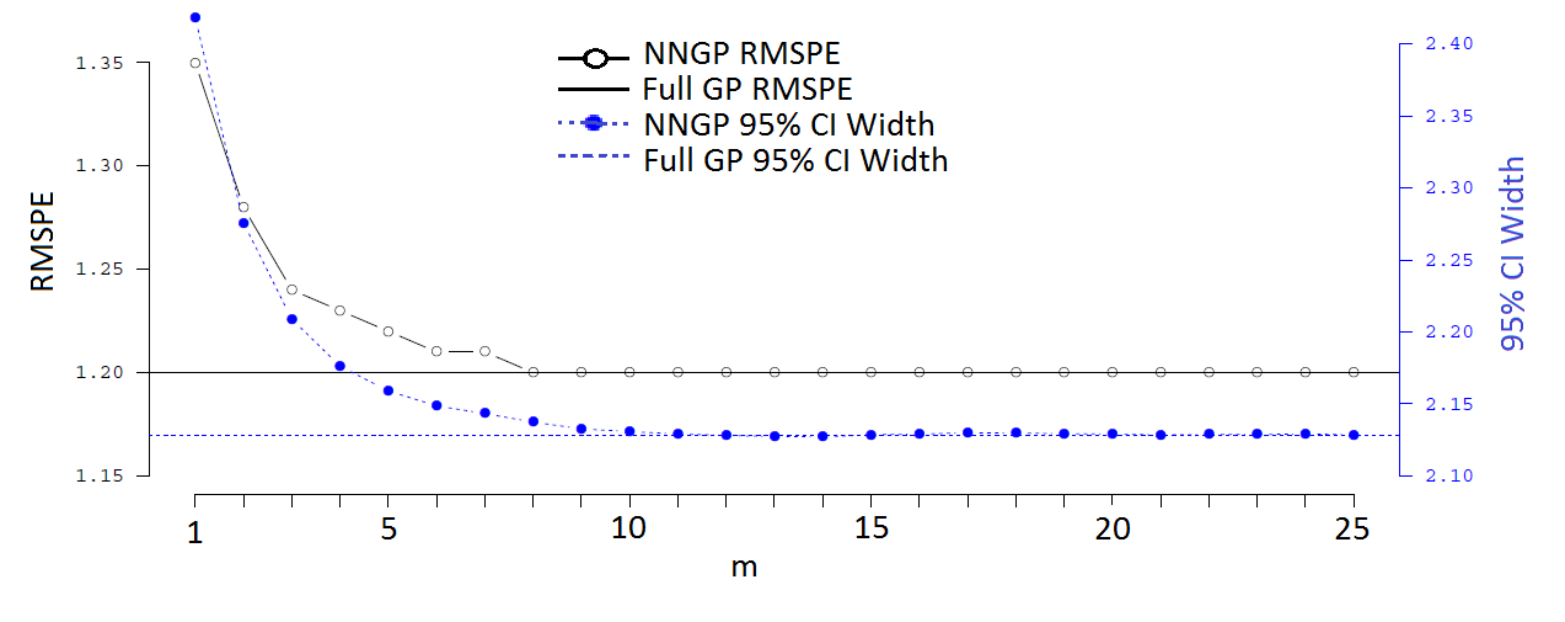
$$N(y \mid X\beta + w, \tau^2 I) \times N(w \mid 0, \tilde{C}(\sigma^2, \phi)) \times N(\beta \mid \mu_\beta, V_\beta)$$
$$\times IG(\tau^2 \mid a_\tau, b_\tau) \times IG(\sigma^2 \mid a_\sigma, b_\sigma) \times Unif(\phi \mid a_\phi, b_\phi)$$

Gibbs sampler:

- 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:

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

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

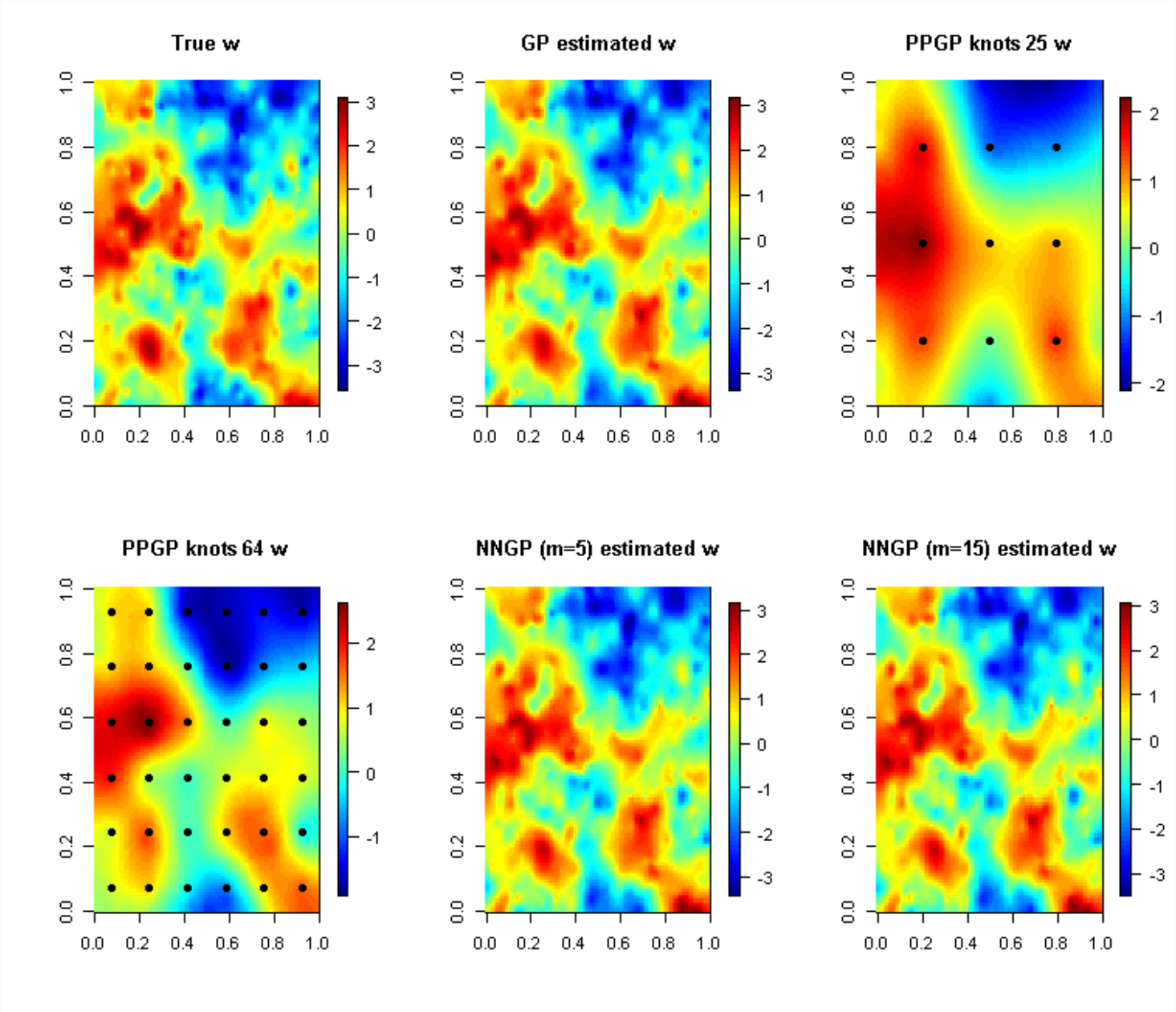
Storage and computation

- 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

spNNGP package in R

- Implements the MCMC for spatial regression model using NNGP
- Full posterior distributions of all parameters available (similar to spBayes)
- Suitable for parallel computing
- Implements NNGP variants like the response model and the MCMC-free conjugate model
- Very suitable for analyzing very large spatial datasets (upto millions of locations)

Predicted surfaces of w



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
- Solution: Back to the marginalized model?

Collapsed NNGP

- Same model:

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

- 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

- $w(s) \sim GP(0, C(\cdot, \cdot \mid \theta)) \Rightarrow y(s) \sim GP(x(s)'\beta, \Sigma(\cdot, \cdot \mid \tau^2, \theta))$
- $\Sigma(s_i, s_j) = C(s_i, s_j \mid \theta) + \tau^2 \, \delta(s_i = s_j)$ (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 $\Rightarrow$ Improved convergence
- Cannot coherently recover $w \mid y$

MCMC convergence

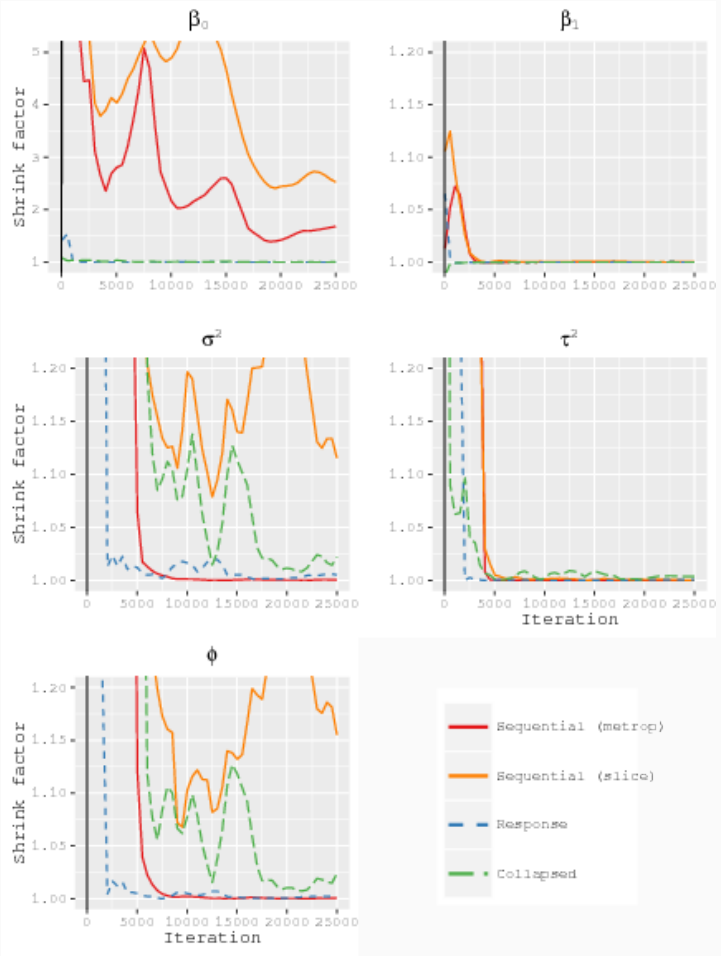
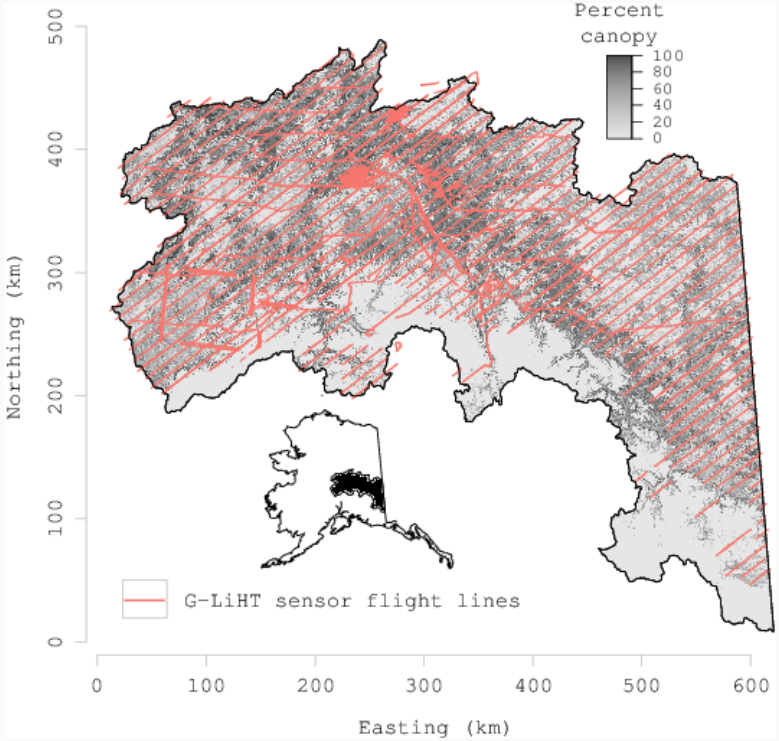
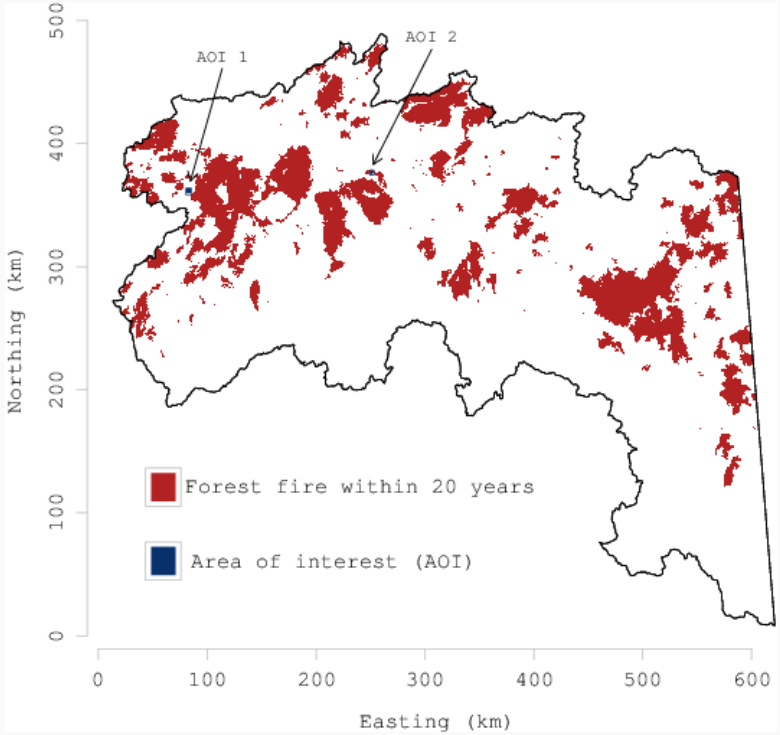


Figure: MCMC convergence diagnostics using Gelman-Rubin shrink factor for different NNGP models for a simulated dataset

Case Study: Alaska Tanana Valley Forest Height Dataset



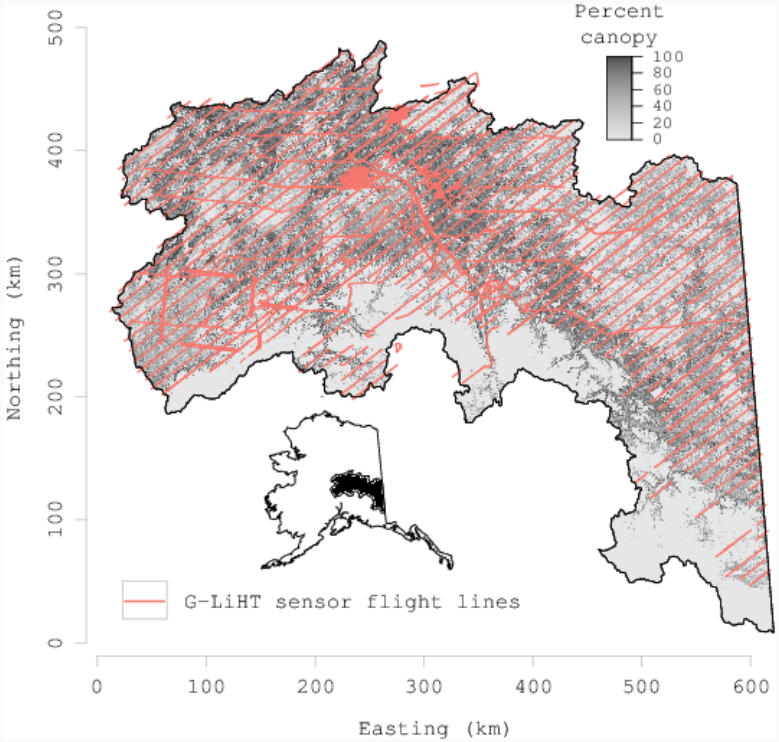


Forest height and tree cover

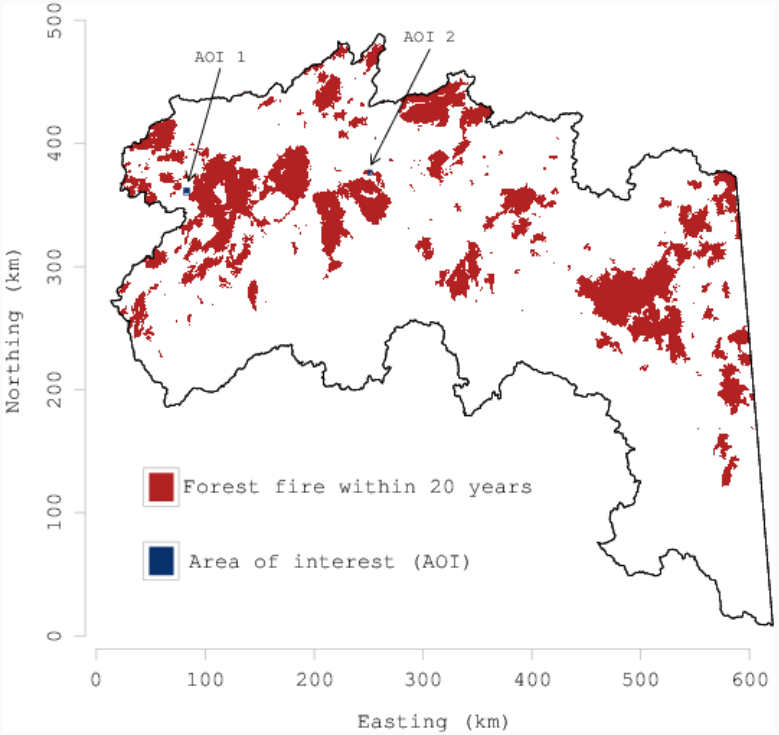
Forest fire history

- Forest height (red lines) data from LiDAR at 5×10^6 locations
- Knowledge of forest height is important for biomass assessment, carbon management etc

Case Study: Alaska Tanana Valley Forest Height Dataset



Forest height and tree cover



Forest fire history

- Goal: High-resolution domainwide prediction maps of forest height
- Covariates: Domainwide tree cover (grey) and forest fire history (red patches) in the last 20 years

Analyzing the data

Models used:

- Non-spatial regression:
  $y_{FH}(s) = \beta_0 + \beta_{tree} x_{tree} + \beta_{fire} x_{fire} + \epsilon(s)$

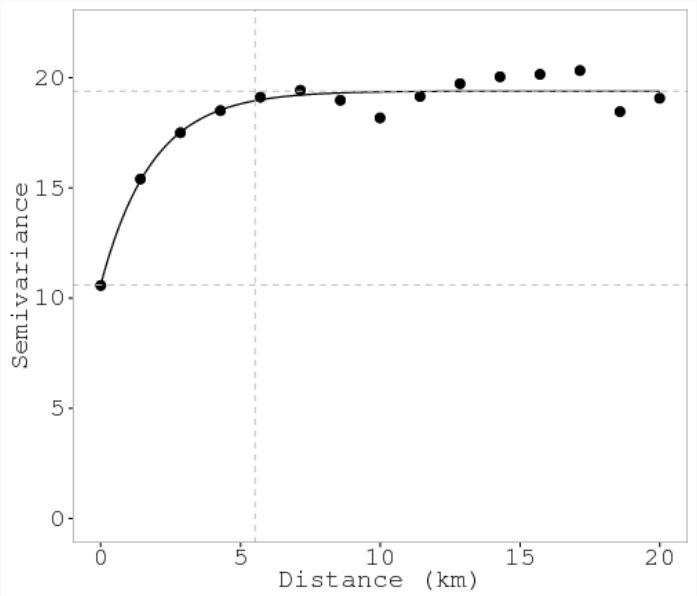


Figure: Variogram of the residuals from non-spatial regression indicates strong spatial pattern

NNGP models

- Collapsed NNGP:
  - $y_{FH}(s) = \beta_0 + \beta_{tree} x_{tree} + \beta_{fire} x_{fire} + w(s) + \epsilon(s)$
  - $w(s) \sim NNGP(0, C(\cdot, \cdot \mid \sigma^2, \phi))$
  - $y_{FH} \sim N(X\beta, \tilde{C} + \tau^2 I)$ where $\tilde{C}$ is the NNGP covariance matrix derived from C
- Response NNGP:
  - $y_{FH}(s) \sim NNGP(\beta_0 + \beta_{tree} x_{tree} + \beta_{fire} x_{fire}, \Sigma(\cdot, \cdot \mid \sigma^2, \phi, \tau^2))$
  - $y_{FH} \sim N(X\beta, \tilde{\Sigma})$ where $\tilde{\Sigma}$ is the NNGP covariance matrix derived from $\Sigma = C + \tau^2 I$

NNGP models

	Non-spatial regression	Collapsed NNGP	Response NNGP
CRPS	2.3	0.86	0.86
RMSPE	4.2	1.73	1.72
CP	93%	94%	94%
CIW	16.3	6.6	6.6

Table: Model comparison metrics for the Tanana valley dataset

- NNGP models perform significantly better than the non-spatial model
- MCMC run time for the NNGP models:
  - Collapsed model: 319 hours
  - Response model: 38 hours
- For massive spatial data, full Bayesian output for even NNGP models require substantial time

Another look at the response model

- Original full GP model: $y(s) \overset{ind}{\sim} N(x(s)'\beta + w(s), \tau^2)$
- $w(s) \sim GP$ with a stationary covariance function $C(\cdot, \cdot \mid \sigma^2, \phi)$
- $Cov(w) = \sigma^2 R(\phi)$
- Full GP model: $y \sim N(X\beta, \Sigma)$ where $\Sigma = \sigma^2 M$
- $M = R(\phi) + \alpha I$
- $\alpha = \tau^2/\sigma^2$ is the ratio of the noise to signal variance
- Response NNGP model: $y \sim N(X\beta, \tilde{\Sigma})$
- $\tilde{\Sigma} = \sigma^2 \tilde{M}$ where $\tilde{M}$ is the NNGP approximation for M

Conjugate NNGP

- $y \sim N(X\beta, \sigma^2 \tilde{M})$
- If ϕ and α are known, M, and hence $\tilde{M}$, are known matrices
- The model becomes a standard Bayesian linear model
- Assume a *Normal Inverse Gamma (NIG)* prior for $(\beta, \sigma^2)'$
- $(\beta, \sigma^2)' \sim NIG(\mu_\beta, V_\beta, a_\sigma, b_\sigma)$, i.e., $\beta \,|\, \sigma^2 \sim N(\mu_\beta, \sigma^2 V_\beta)$ and $\sigma^2 \sim IG(a_\sigma, b_\sigma)$

Conjugate NNGP

- $y \sim N(X\beta, \sigma^2\tilde{M})$, $\tilde{M}$ is known

Joint likelihood:

$$N(y \mid X\beta, \sigma^2\tilde{M}) \times N(\beta \mid \mu_\beta, \sigma^2 V_\beta) \times IG(\sigma^2 \mid a_\sigma, b_\sigma)$$

Conjugate NNGP

- $y \sim N(X\beta, \sigma^2\tilde{M})$, $\tilde{M}$ is known

Joint likelihood:

$$N(y \mid X\beta, \sigma^2\tilde{M}) \times N(\beta \mid \mu_\beta, \sigma^2 V_\beta) \times IG(\sigma^2 \mid a_\sigma, b_\sigma)$$

- Conjugate posterior distribution
  $(\beta, \sigma^2) \mid y \sim NIG(\mu_\beta^*, V_\beta^*, a_\sigma^*, b_\sigma^*)$
- Expressions for μ_β^*, V_β^*, a_σ^* and b_σ^* can be calculated in $O(n)$ time

Conjugate NNGP

- $(\beta, \sigma^2) \,|\, y \sim NIG(\mu^*_\beta, V^*_\beta, a^*_\sigma, b^*_\sigma)$
- Marginal posterior: $\beta \,|\, y \sim MVt_{2a^*_\sigma}(\mu^*_\beta, \frac{b^*_\sigma}{a^*_\sigma} V^*_\beta)$
- $MVt_k(m, V)$ is the *multivariate t* distribution with degrees of k, mean m and scale matrix V
- $E(\beta \,|\, y) = \mu^*_\beta$, $Var(\beta \,|\, y) = \frac{b^*_\sigma}{a^*_\sigma - 1} V^*_\beta$
- Marginal posterior: $\sigma^2 \,|\, y \sim IG(a^*_\sigma, b^*_\sigma)$
- $E(\sigma^2 \,|\, y) = \frac{b^*_\sigma}{a^*_\sigma - 1}$, $Var(\sigma^2 \,|\, y) = \frac{b^{*2}_\sigma}{(a^*_\sigma - 1)^2 (a^*_\sigma - 2)}$
- Exact posterior distributions of β and σ^2 are available

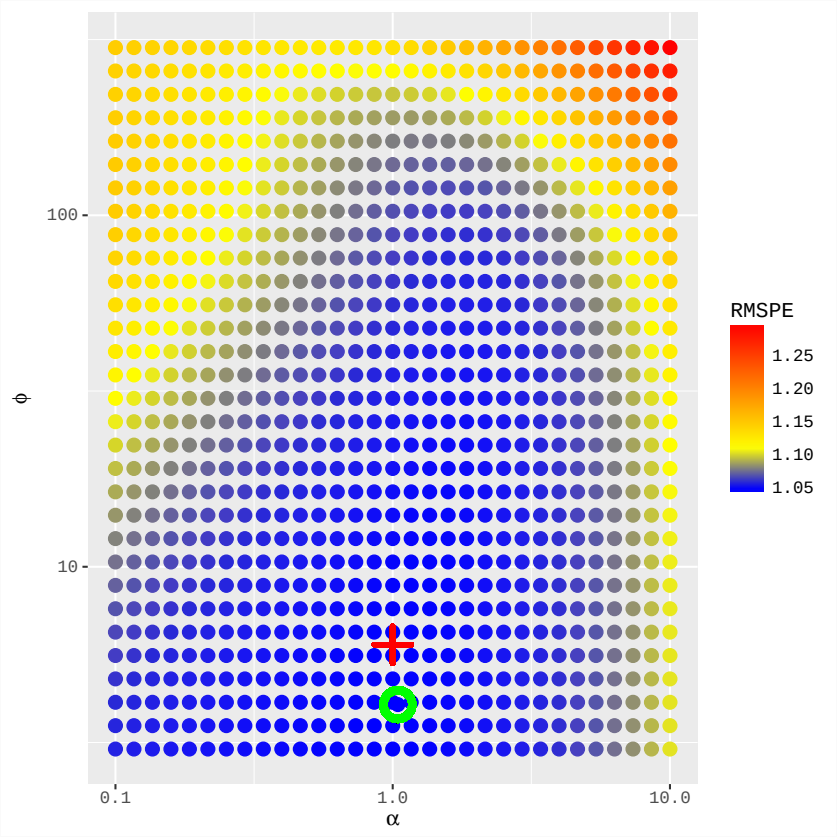
Predictive distributions

- $y(s) \mid y \sim t_{2a_\sigma^*}(m(s), \frac{b_\sigma^*}{a_\sigma^*} v(s))$
- $E(y(s) \mid y) = m(s)$, $Var(y(s) \mid y) = \frac{b_\sigma^*}{a_\sigma^* - 1} v(s)$
- $m(s)$ and $v(s)$ can be computed using $O(m)$ flops
- Exact posterior predictive distributions of $y(s) \mid y$ for any s
- No MCMC required for parameter estimation or prediction

Choosing α and ϕ

- ϕ and α are chosen using K-fold cross validation over a grid of possible values
- Unlike MCMC, cross-validation can be completely parallelized
- Resolution of the grid for ϕ and α can be decided based on computing resources available
- In practice, a reasonably coarse grid often suffices

Choosing α and ϕ



RMSPE

Figure: Simulation experiment: True value ($+$) of (α, ϕ) and estimated value ($\circ$) using 5-fold cross validation

Scalability

- Computation and storage requirements are $O(n)$
- One evaluation time similar to the response NNGP model
- Unlike response NNGP, does not involve any serial MCMC iterations
- For K fold cross validation and G combinations of ϕ and α, total number of evaluations is KG
- Embarassingly parallel: Each of the KG evaluations can proceed in parallel

Scalability

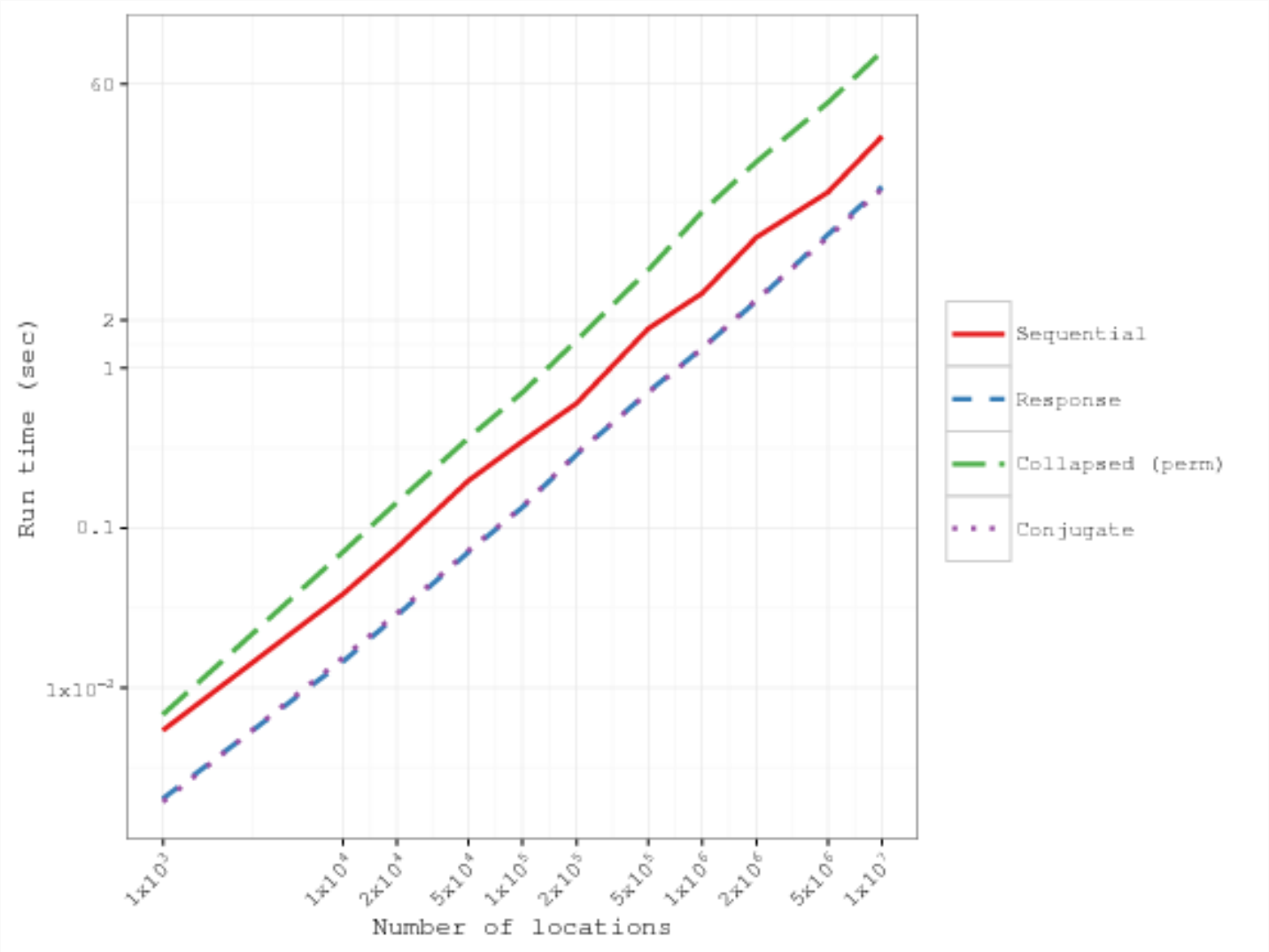


Figure: Run times of different NNGP models with increasing sample size

Comparison of NNGP models

	Sequential	Collapsed	Response	Conjugate
$O(n)$ time	Yes	No	Yes	Yes
Recovery of $w \mid y$	Yes	Yes	No	No
MCMC dimensionality	High	Low	Low	MCMC-free
Fully Bayesian inference	Yes	Yes	Yes	No
Embarassingly parallel	No	No	No	Yes

Alaska Tanana Valley dataset

	Conjugate NNGP	Collapsed NNGP	Response NNGP
β_0	2.51	2.41 (2.35, 2.47)	2.37 (2.31,2.42)
β_{TC}	0.02	0.02 (0.02, 0.02)	0.02 (0.02, 0.02)
β_{Fire}	0.35	0.39 (0.34, 0.43)	0.43 (0.39, 0.48)
σ^2	23.21	18.67 (18.50, 18.81)	17.29 (17.13, 17.41)
τ^2	1.21	1.56 (1.55, 1.56)	1.55 (1.54, 1.55)
ϕ	3.83	3.73 (3.70, 3.77)	4.15 (4.13, 4.19)
CRPS	0.84	0.86	0.86
RMSPE	1.71	1.73	1.72
time (hrs.)	0.002	319	38

Table: Parameter estimates and model comparison metrics for the Tanana valley dataset

- Conjugate model produces estimates and model comparison numbers very similar to the MCMC based NNGP models
- For 5×10^6 locations, conjugate model takes 7 seconds

Multivariate spatial data

- Point-referenced spatial data often come as multivariate measurements at each location.
- Examples:
  - Environmental monitoring: stations yield measurements on ozone, NO, CO, and $PM_{2.5}$.
  - Forestry: measurements of stand characteristics age, total biomass, and average tree diameter.
  - Atmospheric modeling: at a given site we observe surface temperature, precipitation and wind speed
- We anticipate dependence between measurements
  - at a particular location
  - across locations

Multvariate spatial linear model

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

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

- $Cov(w(s_i), w(s_j)) = C(s_i, s_j \mid \theta)$ – a $q \times q$ cross-covariance matrix
- Choices for the function $C(\cdot, \cdot \mid \theta)$
  - 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

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

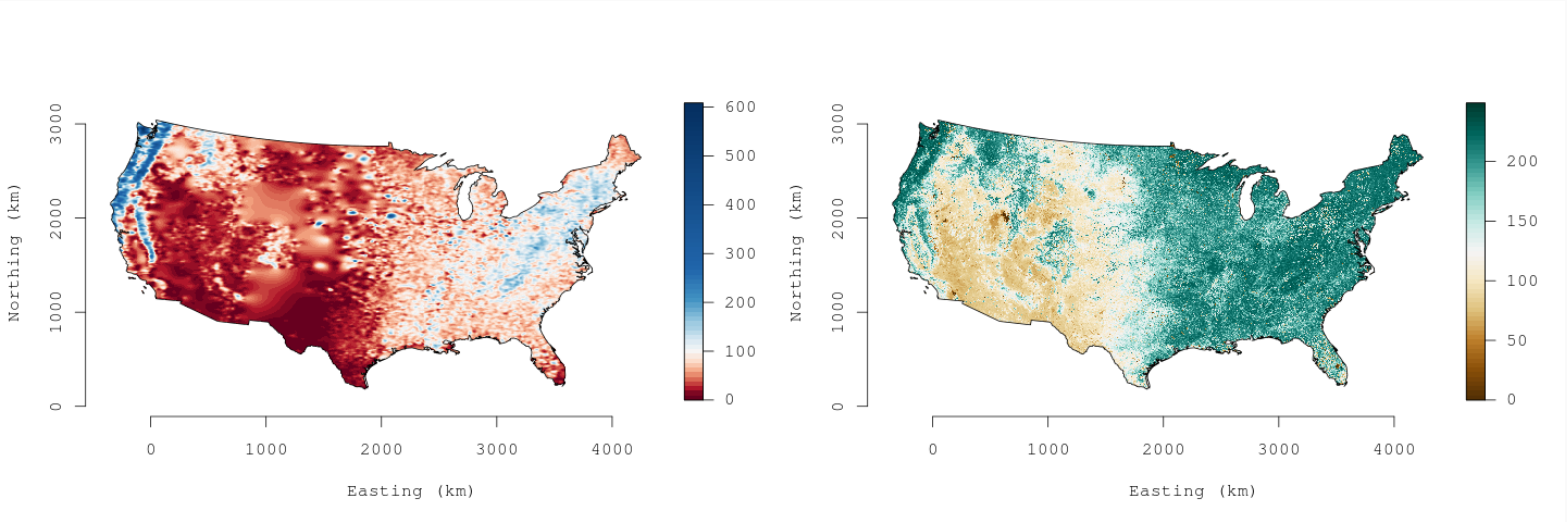
$$\implies w = Aw + \eta; \quad \eta \sim N(0, D), \; D = diag(D_1, D_2, \dots, D_n).$$

- 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



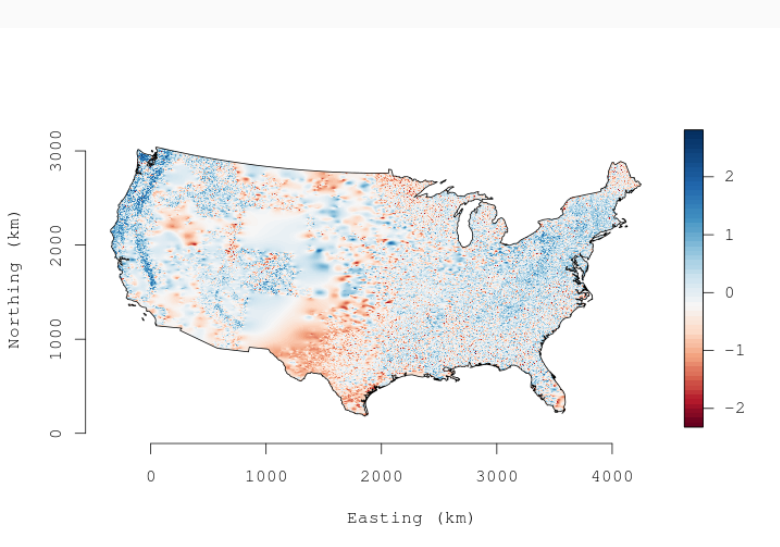
Observed biomass NDVI

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

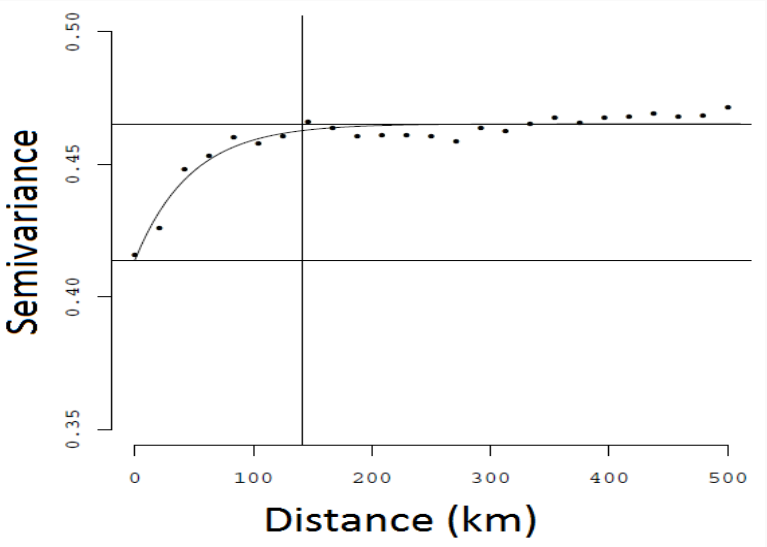
U.S. Forest biomass data

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

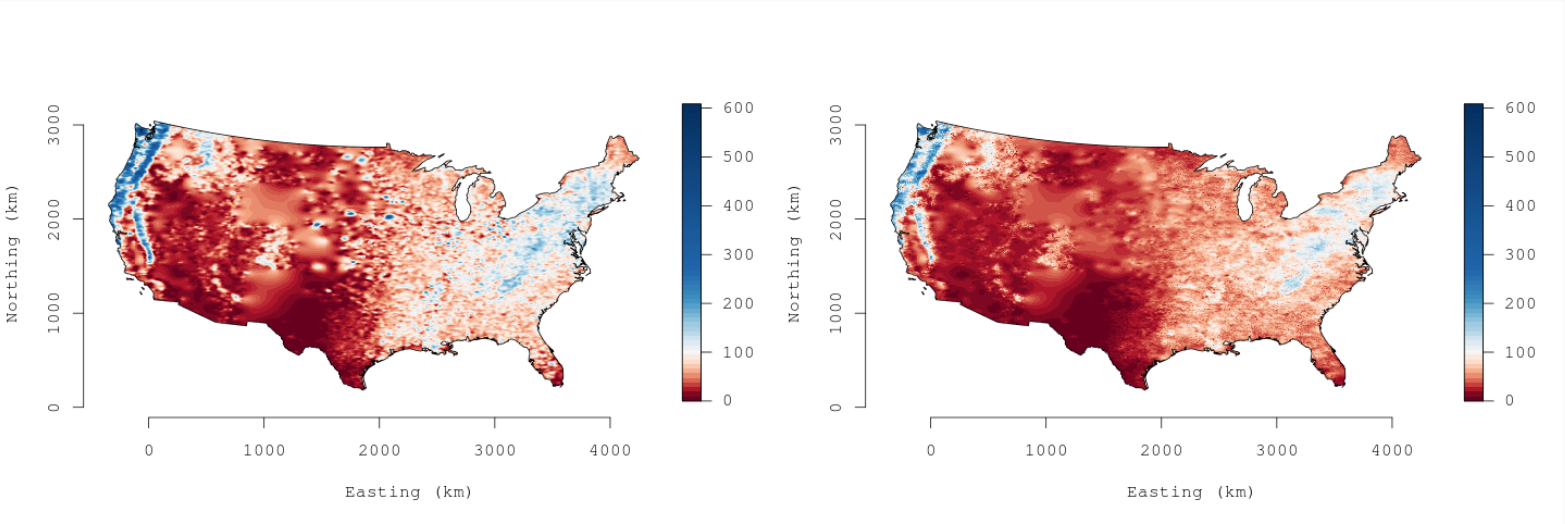
Forest biomass dataset

- $n \approx 10^5$ (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

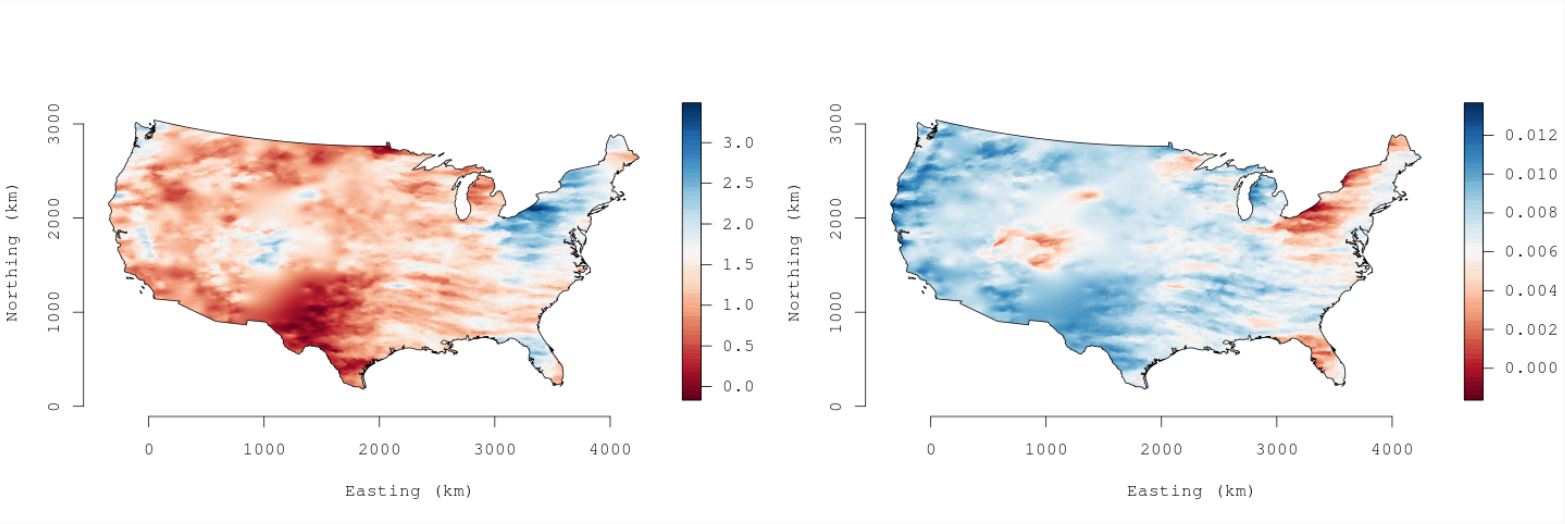
- $Biomass(s) = \beta_0(s) + \beta_1(s) NDVI(s) + \epsilon(s)$
- $w(s) = (\beta_0(s), \beta_1(s))^\top \sim$ Bivariate $NNGP(0, \tilde{C}(\cdot, \cdot \mid \theta))$, $m = 5$
- Time $\approx$ 6 seconds per iteration
- Full inferential output: 41 hours (25000 MCMC iterations)

Forest biomass data



Observed biomass

Fitted biomass



$\beta_0(s)$

$\beta_{NDVI}(s)$

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