Nearest Neighbor Gaussian Processes for Large Spatial Data

Abhi Datta\(^1\), Sudipto Banerjee\(^2\) and Andrew O. Finley\(^3\)

July 31, 2017

\(^1\)Department of Biostatistics, Bloomberg School of Public Health, Johns Hopkins University, Baltimore, Maryland.
\(^2\)Department of Biostatistics, Fielding School of Public Health, University of California, Los Angeles.
\(^3\)Departments of Forestry and Geography, Michigan State University, East Lansing, Michigan.
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
Cons

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
• 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 \( \Rightarrow \)

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

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

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

3
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 \( \Rightarrow \)

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

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

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

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

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

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

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

\begin{align*}
p(y_1)p(y_2 | y_1)p(y_3 | y_1, y_2)p(y_4 | y_1, y_2, y_3) \\
\times p(y_5 | y_1, y_2, y_3, y_4)p(y_6 | y_1, y_2, \ldots, y_5)p(y_7 | y_1, y_2, \ldots, y_6) \ .
\end{align*}
Introducing sparsity via graphical models

\[ p(y_1)p(y_2 | y_1)p(y_3 | y_1, y_2)p(y_4 | y_1, y_2, y_3) \]
\[ p(y_5 | y_1, y_2, y_3, y_4)p(y_6 | y_1, y_2, y_3, y_4, y_5)p(y_7 | y_1, y_2, y_3, y_4, y_5, y_6) \]
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:
  \[
  \mathbb{E}[w_i \mid w_{N(i)}] = \sum_{j \in N(i)} a_{ij}w_j
  \]
- The above linear system is only \( m \times m \)
Choosing neighbor sets

Matern Covariance Function:

\[ C(s_i, s_j) = \frac{1}{2^{\nu-1} \Gamma(\nu)} (|s_i - s_j| \phi)^\nu \mathcal{H}_\nu (|s_i - s_j| \phi); \quad \phi > 0, \nu > 0, \]
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 $A$
- $N(w \mid 0, C) \approx N(w \mid 0, \tilde{C}) ; \tilde{C}^{-1} = (I - A)^\top D^{-1}(I - A)$

\begin{itemize}
  \item $\det(\tilde{C}) = \prod_{i=1}^{n} D_i,$
  \item $\tilde{C}^{-1}$ is sparse with $O(nm^2)$ entries
\end{itemize}
Sparse precision matrices

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

- $\det(\tilde{C}) = \prod_{i=1}^{n} D_i$
- $\tilde{C}^{-1}$ 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 $s \not\in S$, 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**

\[ y(s) = x(s)'\beta + w(s) + \epsilon(s) \]

- \( w(s) \) modeled as NNGP derived from a \( GP(0, C(\cdot, \cdot, | \sigma^2, \phi)) \)
- \( \epsilon(s) \overset{iid}{\sim} N(0, \tau^2) \) contributes to the nugget
- Priors for the parameters \( \beta, \sigma^2, \tau^2 \) and \( \phi \)
- *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 $\beta$, $\tau^2$, $\sigma^2$ and $w(s_i)$’s
- Metropolis step for updating $\phi$
- **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
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 \( \phi \)
- Candidate models: Full GP, Gaussian Predictive Process (GPP) with 64 knots and NNGP
Fitted Surfaces

**Figure:** Univariate synthetic data analysis

- True $w$
- Full GP
- GPP 64 knots

NNGP, $m = 10$

NNGP, $m = 20$
### Parameter estimates

<table>
<thead>
<tr>
<th></th>
<th>True</th>
<th>$m = 10$</th>
<th>$m = 20$</th>
<th>64 knots</th>
<th>Gaussian Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_0$</td>
<td>1</td>
<td>1.00 (0.62, 1.31)</td>
<td>1.03 (0.65, 1.34)</td>
<td>1.30 (0.54, 2.03)</td>
<td>1.03 (0.69, 1.34)</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>5</td>
<td>5.01 (4.99, 5.03)</td>
<td>5.01 (4.99, 5.03)</td>
<td>5.03 (4.99, 5.06)</td>
<td>5.01 (4.99, 5.03)</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>1</td>
<td>0.96 (0.78, 1.23)</td>
<td>0.94 (0.77, 1.20)</td>
<td>1.29 (0.96, 2.00)</td>
<td>0.94 (0.76, 1.23)</td>
</tr>
<tr>
<td>$\tau^2$</td>
<td>0.1</td>
<td>0.10 (0.08, 0.13)</td>
<td>0.10 (0.08, 0.13)</td>
<td>0.08 (0.04, 0.13)</td>
<td>0.10 (0.08, 0.12)</td>
</tr>
<tr>
<td>$\phi$</td>
<td>12</td>
<td>12.93 (9.70, 16.77)</td>
<td>13.36 (9.99, 17.15)</td>
<td><strong>5.61 (3.48, 8.09)</strong></td>
<td>13.52 (9.92, 17.50)</td>
</tr>
</tbody>
</table>
## Model evaluation

<table>
<thead>
<tr>
<th></th>
<th>NNGP $m = 10$</th>
<th>NNGP $m = 20$</th>
<th>Predictive Process 64 knots</th>
<th>Full Gaussian Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIC score</td>
<td>2390</td>
<td>2377</td>
<td><strong>13678</strong></td>
<td>2364</td>
</tr>
<tr>
<td>RMSPE</td>
<td>1.2</td>
<td>1.2</td>
<td><strong>1.68</strong></td>
<td>1.2</td>
</tr>
<tr>
<td>Run time (Minutes)</td>
<td><strong>14.40</strong></td>
<td>46.47</td>
<td>43.36</td>
<td>560.31</td>
</tr>
</tbody>
</table>

- 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
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
• Spatial linear model for $q$-variate spatial data:
  \[ y_i = x_i'(s)\beta_i + w_i(s) + \epsilon_i(s) \text{ for } i = 1, 2, \ldots, q \]

• $\epsilon(s) = (\epsilon_1(s), \epsilon_2(s), \ldots, \epsilon_q(s))' \sim \mathcal{N}(0, E)$ 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

- \( \text{Cov}(w(s_i), w(s_j)) = C(s_i, s_j | \theta) \) – 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 \( C = \text{Cov}(w(s_1), w(s_2), \ldots, w(s_n)) \)

- Not scalable when \( nq \) is large
Multivariate NNGPs

• Cholesky factor approach similar to the univariate case

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

\[\Rightarrow w = Aw + \eta; \quad \eta \sim N(0, D), \quad D = \text{diag}(D_1, D_2, \ldots, D_n).\]

• 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

- 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

\[ \text{Biomass} = \beta_0 + \beta_1 \text{NDVI} + \text{error}, \quad \hat{\beta}_0 = 1.043, \quad \hat{\beta}_1 = 0.0093 \]

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

- $\text{Biomass}(s) = \beta_0(s) + \beta_1(s)\text{NDVI}(s) + \epsilon(s)$

- $w(s) = (\beta_0(s), \beta_1(s))^\top \sim \text{Bivariate NNGP}(0, \tilde{C}(\cdot, \cdot | \theta)), m = 5$

- 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:
  \[ y(s) = x(s)' \beta + w(s) + \epsilon(s) \]
  \[ w(s) \sim \text{NNGP}(0, C(\cdot, \cdot | \theta)) \]
  \[ \epsilon(s) \sim \text{iid } \mathcal{N}(0, \tau^2) \]

- Vector form \( y \sim \mathcal{N}(X \beta + w, \tau^2 I); w \sim \mathcal{N}(0, \tilde{C}(\theta)) \)

- **Collapsed model:** Marginalizing out \( w \), we have
  \[ y \sim \mathcal{N}(X \beta, \tau^2 I + \tilde{C}(\theta)) \]
Collapsed NNGP

Model

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

- Only involves few parameters \( \beta, \tau^2 \) 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

- $w(s) \sim GP(0, C(\cdot, \cdot | \theta)) \Rightarrow y(s) \sim GP(x(s)'\beta, \Sigma(\cdot, \cdot | \tau^2, \theta))$
- $\Sigma(s_i, s_j) = C(s_i, s_j | \theta) + \tau^2 \delta(s_i = s_j)$ ($\delta$ 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 $w | y$
## Comparison of NNGP models

<table>
<thead>
<tr>
<th></th>
<th>Sequential</th>
<th>Collapsed</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(n)$ time</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Recovery of $w \mid y$</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Parameter dimensionality</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>
Comparison of NNGP models

**Figure:** Run time per iteration as a function of number of locations for different NNGP models
Comparison of NNGP models

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