Nearest Neighbor Gaussian Process (NNGP)

By: Mariya Harris, Vaidehi Dixit, and Enrique Pena
STAT 433/533 Applied Spatial Statistics
Midterm 2
10/14/2020
Full Gaussian Process (GP)

**Spatial linear mixed effects** :

\[ y(s_i) = X(s_i)^T \beta + w(s_i) + \varepsilon(s_i) \]  

(1)

\( \beta \) : regression coefficients,  
\( w(s) \) : random spatial effect at a specific site,  
\( \varepsilon \) : non-spatial random noise

\( w \) follows a **zero-mean multivariate Gaussian distribution** with covariance matrix \( C(\theta) \), and \( \varepsilon \) consists of iid Gaussian with mean 0 and variance \( \tau^2 \).

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

(2)

**Frequentist approach** : Maximize the likelihood of \( y \) with respect to \( \beta, \tau^2, \) and \( \theta \)

**Bayesian framework** : Assign priors to parameters in eq. 2 to obtain posterior inferences via

Markov chain Monte Carlo (MCMC)
Motivation

The Full GP is **computationally expensive** for large datasets (e.g., LandSAT data)

- Inverting the dense $n \times n$ covariance matrix involves $O(n^2)$ storage and $O(n^3)$ computations

- **Solution**: It is better to deal with a **low rank** model or a **sparse covariance matrix**

- **Nearest Neighbors Gaussian Process (NNGP)** is one such method which uses a sparse covariance matrix to analyze large spatial datasets.
Nearest Neighbor Gaussian Process (NNGP)

Sparsity is introduced by specifying a conditional joint distribution in the spatial random effect, $w(s)$, where,

$$w(s_i) | w_{1:(i-1)} = C(s_1, s_{1:(i-1)}) \Sigma_{1:(i-1)}^{-1} w_{1:(s-1)} + \eta(s_i)$$ (3)

$w_{1:(i-1)}$ is replaced by a smaller set of $m$ nearest neighbors of $s_i$

$\Sigma_{1:(i-1)}^{-1}$ is the covariance matrix from the previous sites

$\eta$'s are independent Gaussian with mean zero

Collectively, $w$ can be expressed as,

$$w = Aw + \eta$$ (4)

Where, $A$ is a lower triangular matrix with at most $m$, non-zero entries in each row
Nearest Neighbor Gaussian Process (NNGP)

\[ w \sim N(0, C(\theta)) \]  

(5)

- NNGP constructs a sparse covariance matrix \( C(\theta)^{-1} \) and evaluates the likelihood of (4) using only \( O(n^1) \) storage. The new sparse model is,

\[ y \sim N(X\beta, \Sigma(\phi)) \]  

(6)

Where,

\( \Sigma(\phi) \) is the sparse covariance matrix derived from the full GP model.
Nearest Neighbor Gaussian Process (NNGP)

For $M = 5$...

General Equation: $w(s_i) | w_{1:(i-1)} = C(s_1, s_{1:(i-1)}) * \sum_{1:(i-1)} w_{1:(s-1)} + \eta(s_i)$

$$w_1 = \begin{bmatrix} 0 & 0.8 & 0 & 0.2 & 0.01 \\ 0.5 & 0.7 & 0.9 & 0.3 & 0.6 \\ 0.2 & 0.6 & 0.4 & 0.6 & 0.5 \end{bmatrix}$$

$$w = Aw + \eta$$
Goal and Objectives

Goal
• Implement the NNGP method (latent or response or conjugate) and compare results to classical likelihood technique, i.e., MLE
• Comparison in terms of time efficiency and prediction performance

Procedure
• Pick a covariance model for fitting a spatial model to the given dataset
• For LandSAT-generated data, we pick the exponential covariance model
• Use the latent NNGP (Bayesian) model and MLE to estimate the model parameters $\beta$, $\sigma_w^2$, $\tau^2$, and $\phi$
• Predict the response $Y(s)$ at 1000 test locations using equation 9

\[ w(s) \mid w_{1:n} = a'(s)w_{1:n} + \eta(s) \]  \hspace{1cm} (7)
Implementation – Basic setup

• Data: LandSAT (n = 937,208)

• Data was subset to 21 groups of 1000 observations each: 20 training groups and 1 testing group

• NA values were removed after sub-setting data

• R package: “spNNGP” (Finley, Datta, Banerjee (2020))
Implementation – setup for NNGP

- **Latent NNGP** was the specific method used, which is based on a Bayesian approach

- R Function used: `spNNGP()`

- Decide priors for the parameters

- Theoretically, max number of neighbors = 25 should give a good approximation

- For a Bayesian approach, convergence of parameter estimates is crucial

- Performance is affected by number of neighbors, number of MCMC iterations

- Number of Neighbors used: 5, 10, and 15
  - Intended Number of MCMC iterations: 30,000
  - Intended Burned iterations: 5000

- Final MCMC iterations = 5000, burned = 1000
A for loop was ran for val = 20 iterations:
- Loop 1 used g=1 as training
- Loop 2 used g = 1,2 as training
- Loop n used g = 1,...,n as training
Results: Run-time for NNGP and MLE

**Time Plot**

- **Observations** vs. **Time (seconds)**
- Variables: nngp_time_5, nngp_time_10, nngp_time_15

**Coverage Plot**

- **Coverage** vs. **Observations**
- Variables: COVER, COVER_10, COVER_15
<table>
<thead>
<tr>
<th>Parameter</th>
<th>5 neighbors</th>
<th>10 neighbors</th>
<th>15 neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_0$</td>
<td><img src="image" alt="Trace Plot" /></td>
<td><img src="image" alt="Trace Plot" /></td>
<td><img src="image" alt="Trace Plot" /></td>
</tr>
<tr>
<td>$\beta_1$</td>
<td><img src="image" alt="Trace Plot" /></td>
<td><img src="image" alt="Trace Plot" /></td>
<td><img src="image" alt="Trace Plot" /></td>
</tr>
<tr>
<td>$\beta_2$</td>
<td><img src="image" alt="Trace Plot" /></td>
<td><img src="image" alt="Trace Plot" /></td>
<td><img src="image" alt="Trace Plot" /></td>
</tr>
<tr>
<td>$\sigma^2_w$</td>
<td><img src="image" alt="Trace Plot" /></td>
<td><img src="image" alt="Trace Plot" /></td>
<td><img src="image" alt="Trace Plot" /></td>
</tr>
<tr>
<td>$\tau^2$</td>
<td><img src="image" alt="Trace Plot" /></td>
<td><img src="image" alt="Trace Plot" /></td>
<td><img src="image" alt="Trace Plot" /></td>
</tr>
<tr>
<td>$\phi$</td>
<td><img src="image" alt="Trace Plot" /></td>
<td><img src="image" alt="Trace Plot" /></td>
<td><img src="image" alt="Trace Plot" /></td>
</tr>
</tbody>
</table>
### Results: Parameter estimates

<table>
<thead>
<tr>
<th>Parameter (neighbors = 5)</th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\sigma_w^2$</th>
<th>$\tau^2$</th>
<th>$\phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate (se)</td>
<td>75.47(37.00)</td>
<td>0.25(0.32)</td>
<td>-1.23(0.19)</td>
<td>28.91(11.38)</td>
<td>0.006(0.0001)</td>
<td>0.091 (0.036)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter (neighbors = 10)</th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\sigma_w^2$</th>
<th>$\tau^2$</th>
<th>$\phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate (se)</td>
<td>-35.65 (187.68)</td>
<td>-0.09(0.07)</td>
<td>0.65(4.66)</td>
<td>25.89(9.03)</td>
<td>0.0005(0.00004)</td>
<td>0.09(0.02)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter (neighbors = 15)</th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\sigma_w^2$</th>
<th>$\tau^2$</th>
<th>$\phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate (se)</td>
<td>-155.38 (190.58)</td>
<td>-0.11 (0.10)</td>
<td>3.48 (4.82)</td>
<td>33.65 (6.49)</td>
<td>0.0006 (0.00001)</td>
<td>0.06 (0.01)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter (mle)</th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\sigma_w^2$</th>
<th>$\tau^2$</th>
<th>$\phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate (se)</td>
<td>199.34 (126.94)</td>
<td>-0.03(0.09)</td>
<td>4.72(3.18)</td>
<td>0.01</td>
<td>0</td>
<td>0.003</td>
</tr>
</tbody>
</table>
Prediction performance for NNGP (10 neighbors)
Prediction performance for MLE