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 :

$$y(\boldsymbol{s}_i) = X(\boldsymbol{s}_i)^T \boldsymbol{\beta} + w(\boldsymbol{s}_i) + \varepsilon(\boldsymbol{s}_i) \qquad (1)$$

$\boldsymbol{\beta}$: regression coefficients,

$\boldsymbol{w}(\boldsymbol{s})$: random spatial effect at a specific site, $\boldsymbol{\varepsilon}$: non-spatial random noise

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

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

Frequentist approach : Maximize the likelihood of y with respect to β, τ^2, and $\boldsymbol{\theta}$

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* $\boldsymbol{n\ x\ n}$ *covariance matrix involves* $\boldsymbol{O}(\boldsymbol{n^2})$ *storage and* $\boldsymbol{O}(\boldsymbol{n^3})$ *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,

$$w(s_i)|w_{1:(i-1)} = C\left(\boldsymbol{s}_1, \boldsymbol{s}_{1:(i-1)}\right)\Sigma_{1:(i-1)}^{-1} w_{1:(s-1)} + \boldsymbol{\eta}(s_i) \quad (3)$$

$w_{1:(i-1)}$ 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,

$$w = \boldsymbol{A}w + \boldsymbol{\eta} \quad (4)$$

Where, **A** is a lower triangular matrix with at most m, non-zero entries in each row

Nearest Neighbor Gaussian Process (NNGP)

$$w \sim N(0, C(\boldsymbol{\theta})) \quad (5)$$

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

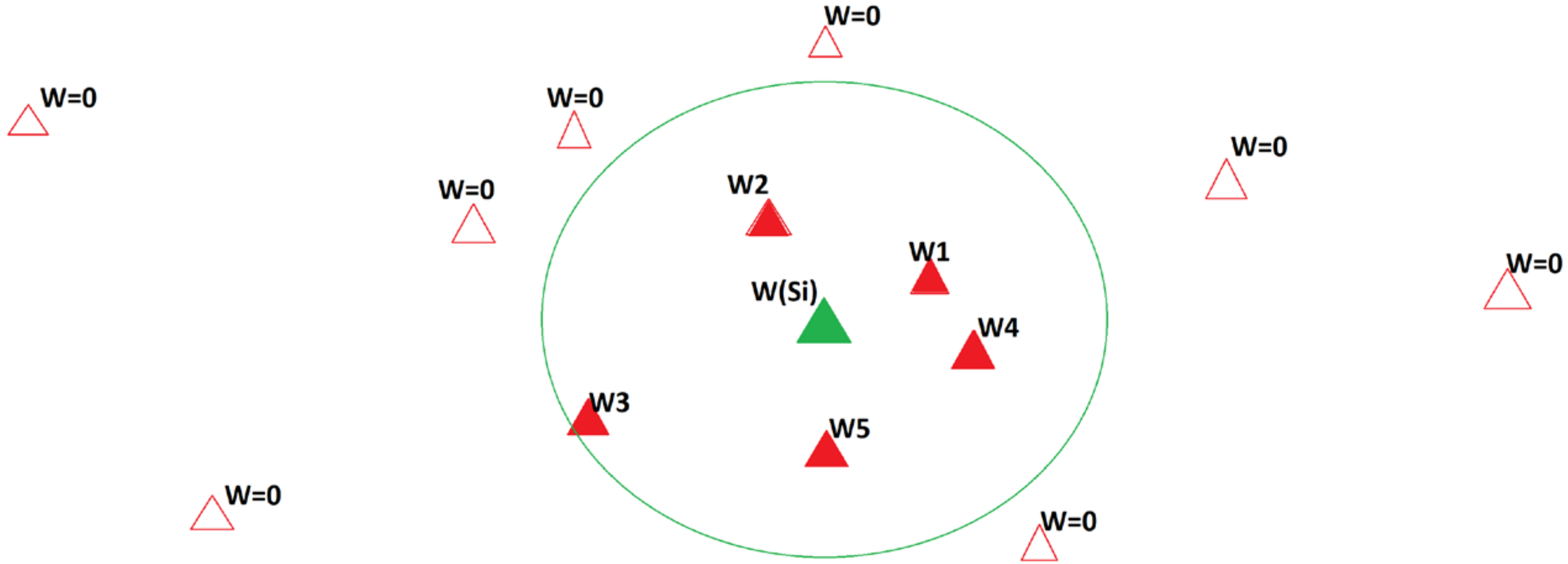
$$y \sim N(X\beta, \sum(\phi)) \quad (6)$$

Where,

$\sum(\phi)$ is the sparse covariance matrix derived from the full GP model

Nearest Neighbor Gaussian Process (NNGP)

For M = 5…



General Equation: $w(s_i)|w_{1:(i-1)} = C\left(\boldsymbol{s}_1, \boldsymbol{s}_{1:(i-1)}\right) * \Sigma^{-1}_{1:(i-1)} * w_{1:(s-1)} + \boldsymbol{\eta}(s_i)$

		W1:1	W1:2	W1:3	W1:4	W1:5				
	W1:1	0						0.2		
	W1:2	0.8	0					0.4		
W(Si) =	W1:3	0.5	0.7	0			*	0.6	+	$\boldsymbol{\eta}(s_i)$
	W1:4	0.2	0.6	0.9	0			0.3		
	W1:5	0.01	0.3	0.4	0.6	0		0.5		

$w = \boldsymbol{A}w + \boldsymbol{\eta}$ → A * w + $\boldsymbol{\eta}(s_i)$

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 β, σ_w^2, τ^2, and ϕ
- Predict the response Y(s) at 1000 test locations using equation 9

$$w(s) \mid w_{1:n} = a'(s) w_{1:n} + \boldsymbol{\eta}(s) \tag{7}$$

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

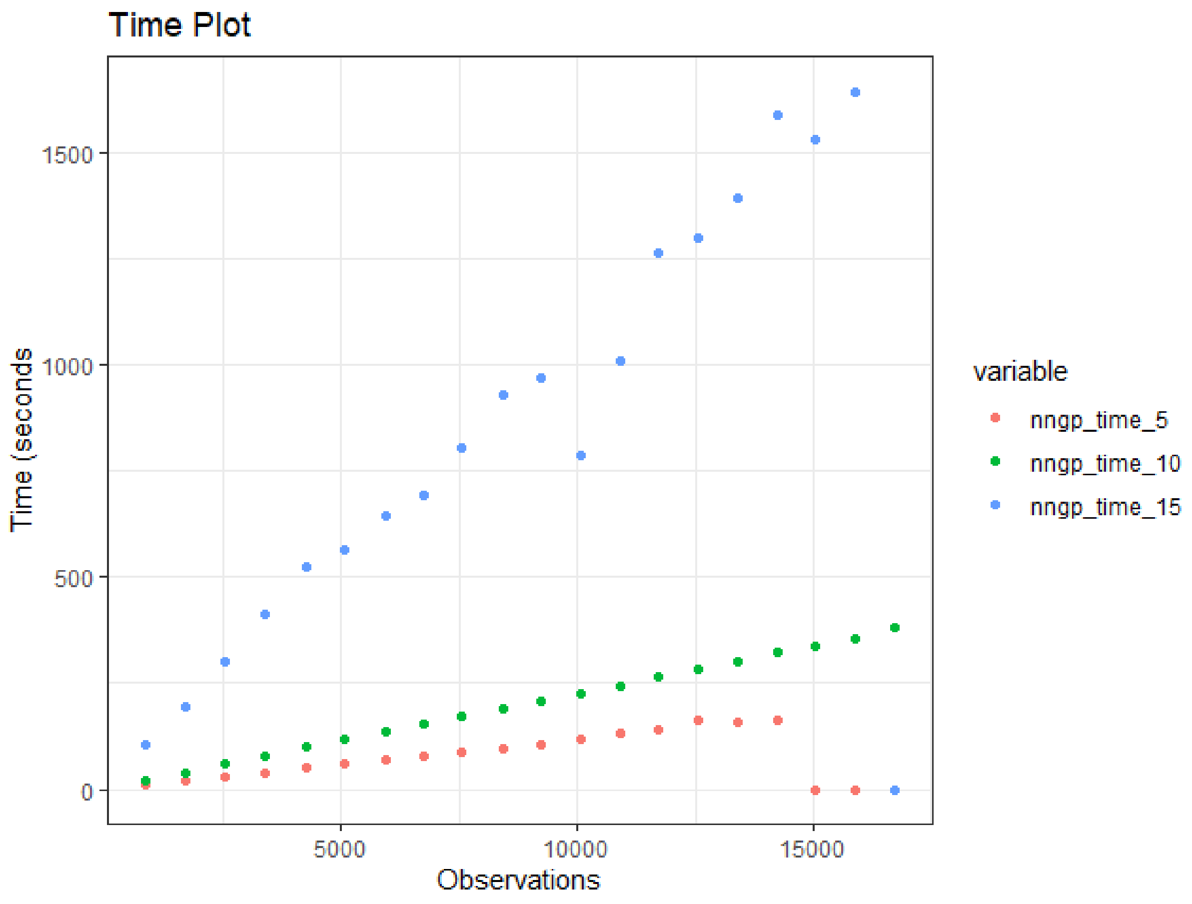
Implementation – R code snippet

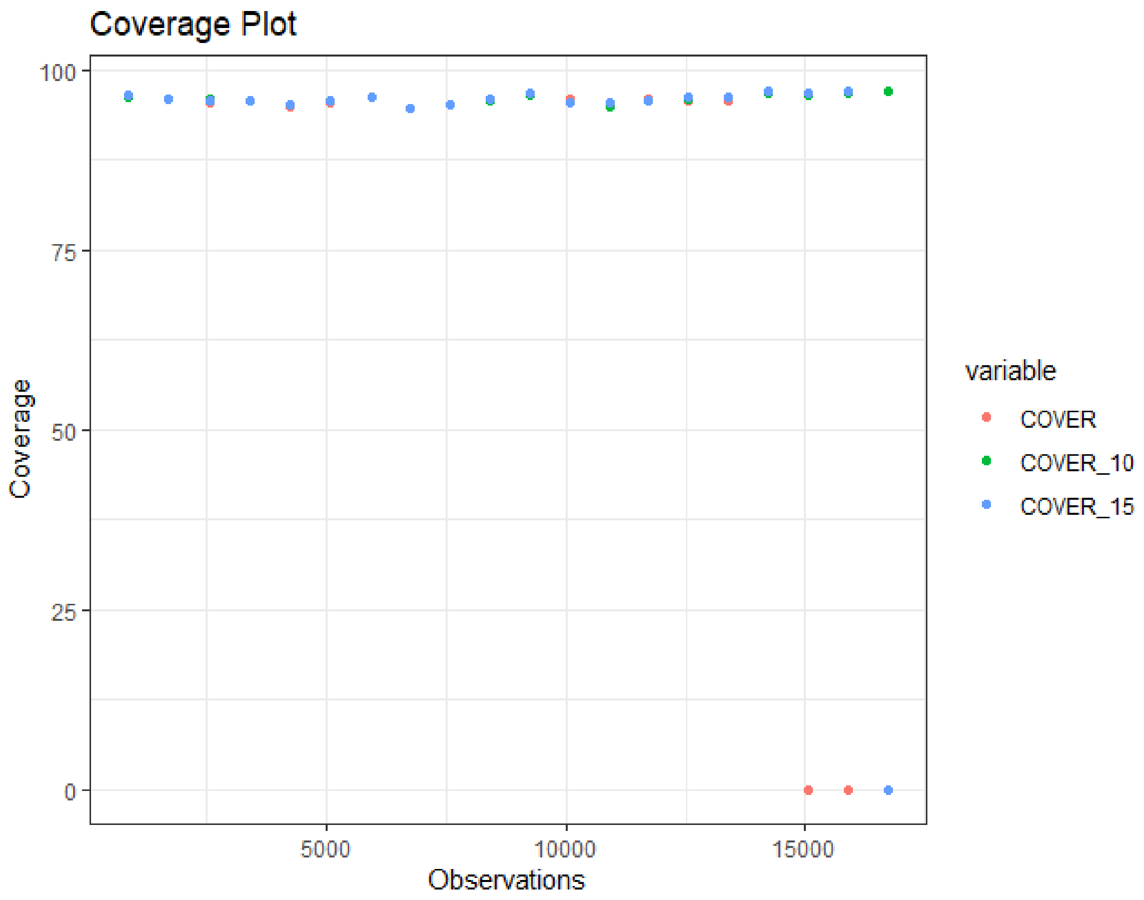
```r
# Priors
n.samples <- 5000
starting <- list("phi"=0.05, "sigma.sq"=1, "tau.sq"=1)
priors <- list("phi.Unif"=c(0.05, 1/0.1), "sigma.sq.IG"=c(2, 1), "tau.sq.IG"=c(2, 1))
cov.model <- "exponential"
tuning <- list("phi"=0.2)

tick = Sys.time()

# Latent NNGP model
sim.s[[val]] <- spNNGP(formula=y_total~X_total,
                       coords=X_total,
                       starting=starting,
                       tuning=tuning,
                       priors=priors,
                       cov.model=cov.model,
                       n.samples=n.samples,
                       n.neighbors=5,
                       method="latent",
                       n.omp.threads=4,
                       n.report=1000,
                       fit.rep=TRUE,
                       sub.sample=list(start=1000),
                       return.neighbor.info = TRUE)



sim.p[[val]] <- predict(sim.s[[val]],
                        X.0=cbind(1,x.ho),
                        coords.0=coords.ho,
                        sub.sample=list(start=1000, thin=10),
                        n.omp.threads = 4,
                        n.report=1000)

tock = Sys.time()
nngp_time_5[val] = difftime(tock, tick, units = "secs")
```

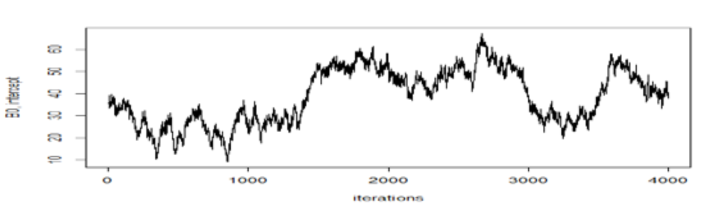
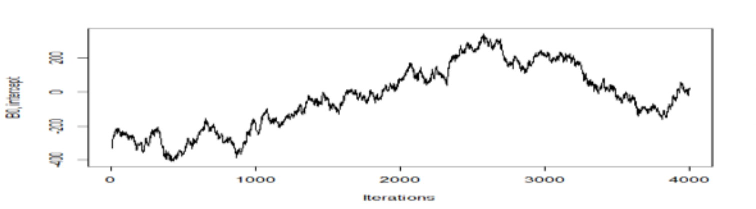
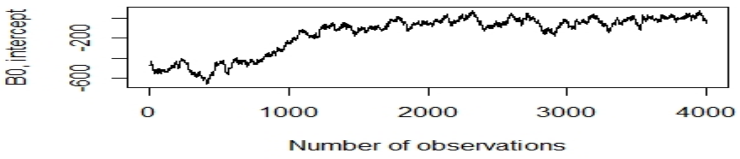
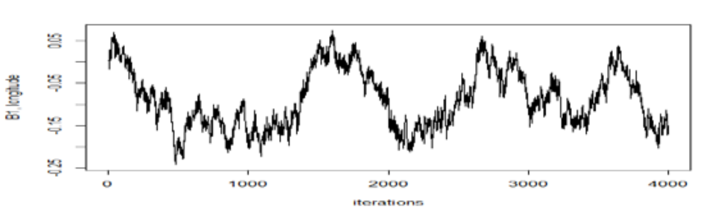
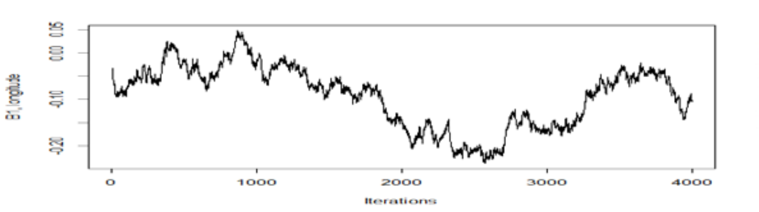
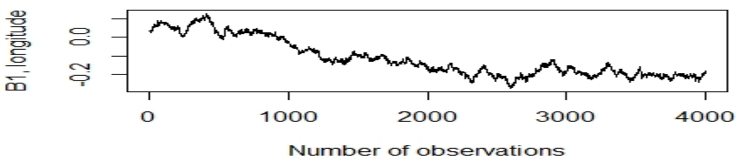
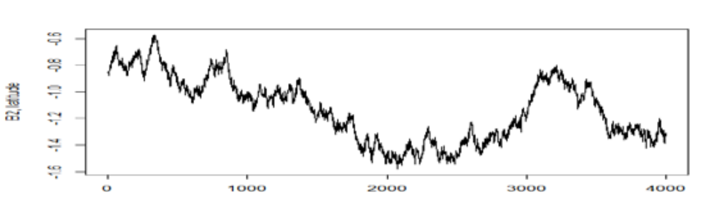
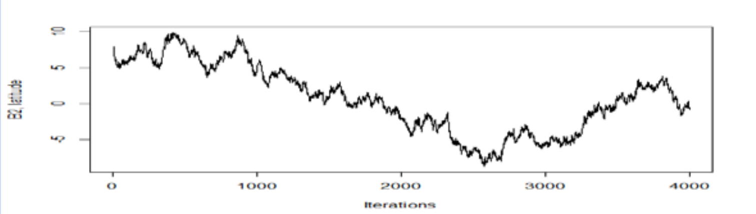
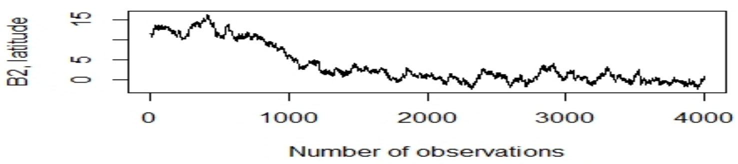
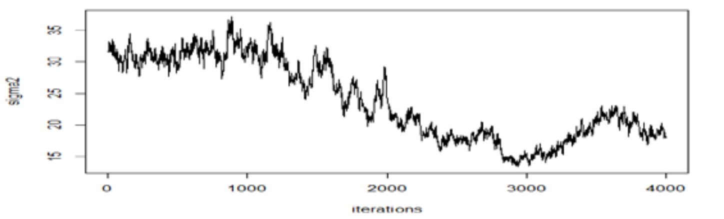
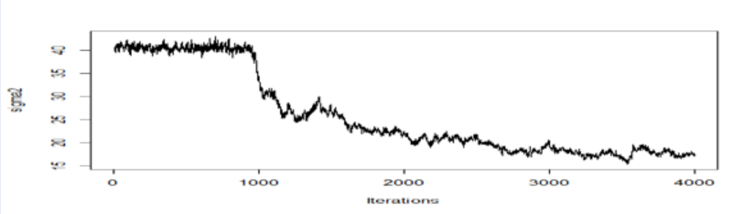
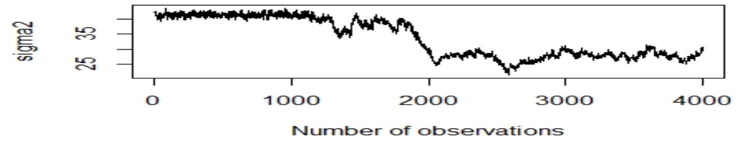
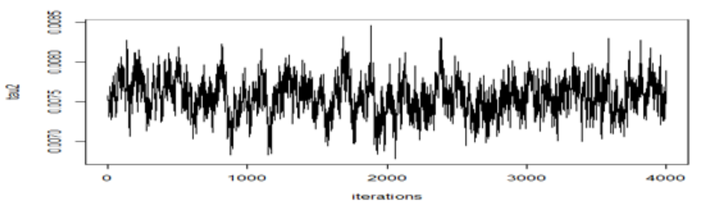
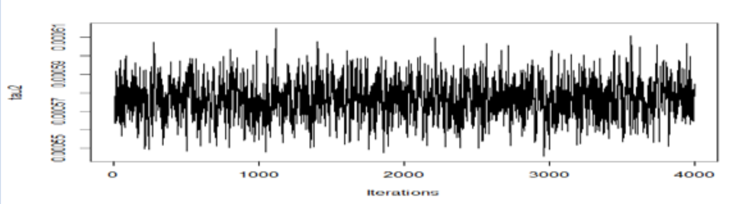
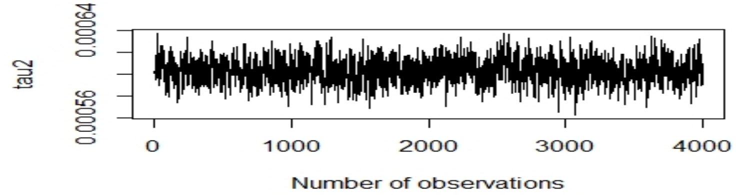
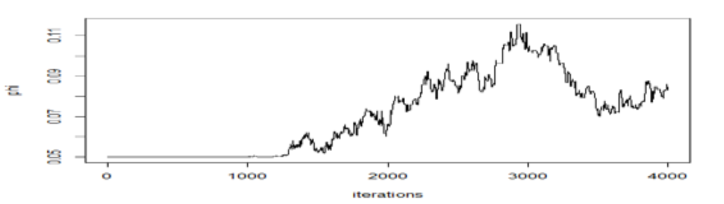
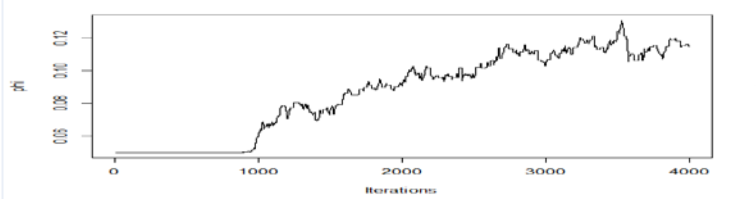
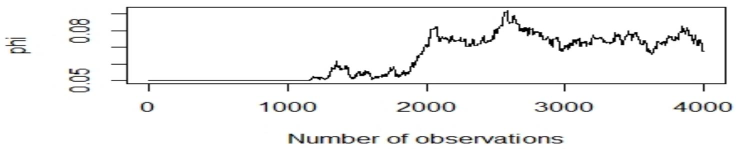
A for loop was ran for *val* = 20 iterations:

- 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





<table>
<tr><th>Parameter</th><th>5 neighbors</th><th>10 neighbors</th><th>15 neighbors</th></tr>
<tr><td>β_0</td><td></td><td></td><td></td></tr>
<tr><td>β_1</td><td></td><td></td><td></td></tr>
<tr><td>β_2</td><td></td><td></td><td></td></tr>
<tr><td>σ_w^2</td><td></td><td></td><td></td></tr>
<tr><td>τ^2</td><td></td><td></td><td></td></tr>
<tr><td>ϕ</td><td></td><td></td><td></td></tr>
</table>

Results : Parameter estimates

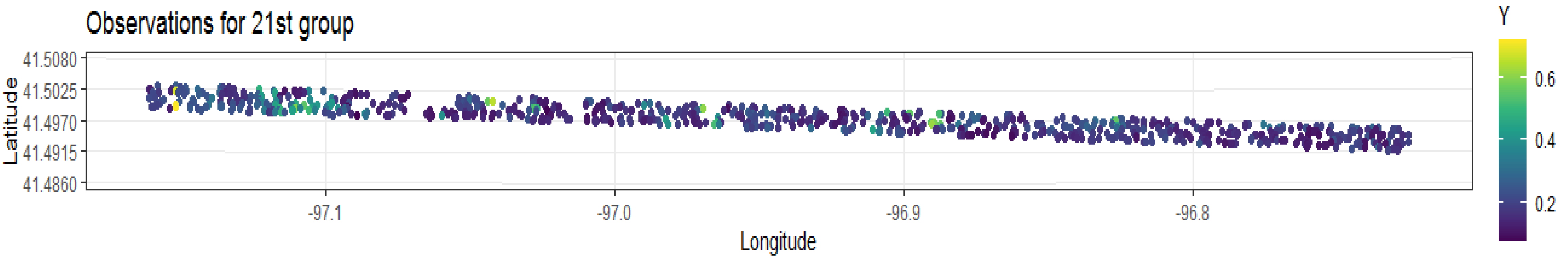
Parameter (neighbors = 5)	β_0	β_1	β_2	σ_w^2	τ^2	ϕ
Estimate(se)	75.47(37.00)	0.25(0.32)	-1.23(0.19)	28.91(11.38)	0.006(0.0001)	0.091 (0.036)

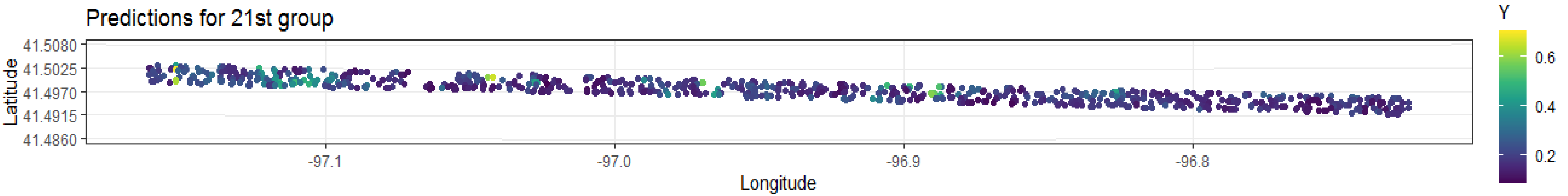
Parameter (neighbors = 10)	β_0	β_1	β_2	σ_w^2	τ^2	ϕ
Estimate(se)	-35.65 (187.68)	-0.09(0.07)	0.65(4.66)	25.89(9.03)	0.0005(0.00004)	0.09(0.02)

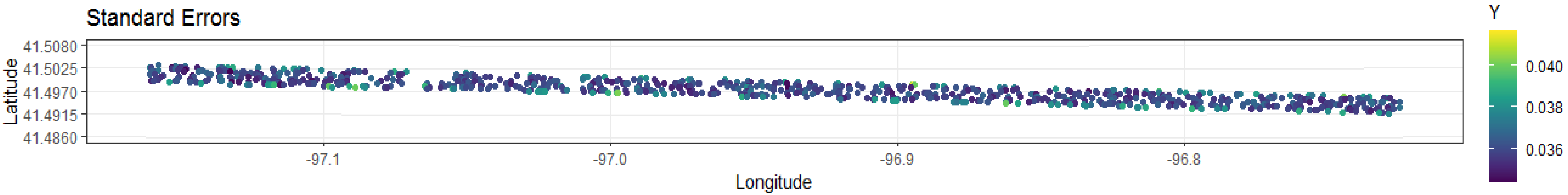
Parameter (neighbors = 15)	β_0	β_1	β_2	σ_w^2	τ^2	ϕ
Estimate(se)	-155.38 (190.58)	-0.11 (0.10)	3.48 (4.82)	33.65(6.49)	0.0006 (0.00001)	0.06 (0.01)

Parameter (mle)	β_0	β_1	β_2	σ_w^2	τ^2	ϕ
Estimate(se)	199.34 (126.94)	-0.03(0.09)	4.72(3.18)	0.01	0	0.003

Prediction performance for NNGP(10 neighbors)







Prediction performance for MLE