Introduction to Geostatistics

Andrew Finley[1] & Jeffrey Doser[2]

May 15, 2023

[1]Department of Forestry, Michigan State University.

[2]Department of Integrative Biology, Michigan State University.

- Course materials available at https://doserjef.github.io/CASANR23-Spatial-Modeling/

What is spatial data?

- Any data with some geographical information (i.e., spatially indexed)
- Common sources of spatial data: agricultural, climatology, forestry, ecology, environmental health, disease epidemiology, product marketing, etc.
  - have many important predictors and response variables
  - are often presented as maps

What is spatial data?

- Any data with some geographical information (i.e., spatially indexed)
- Common sources of spatial data: agricultural, climatology, forestry, ecology, environmental health, disease epidemiology, product marketing, etc.
  - have many important predictors and response variables
  - are often presented as maps
- Other examples where spatial need not refer to space on earth:
  - Genetics (position along a chromosome)
  - Neuroimaging (data for each voxel in the brain)

Point-referenced spatial data

- Each observation is associated with a location (point)
- Data represents a sample from a continuous spatial domain
- Also referred to as geocoded or geostatistical data

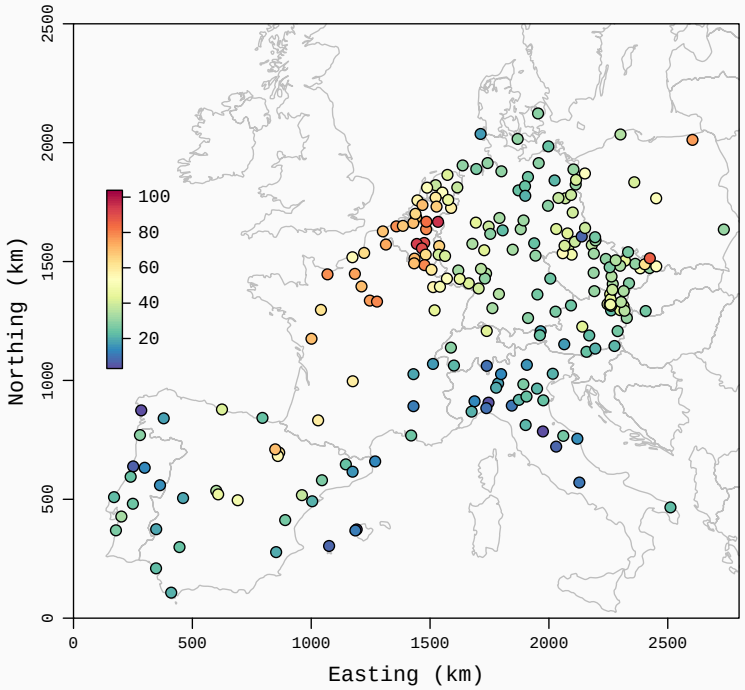


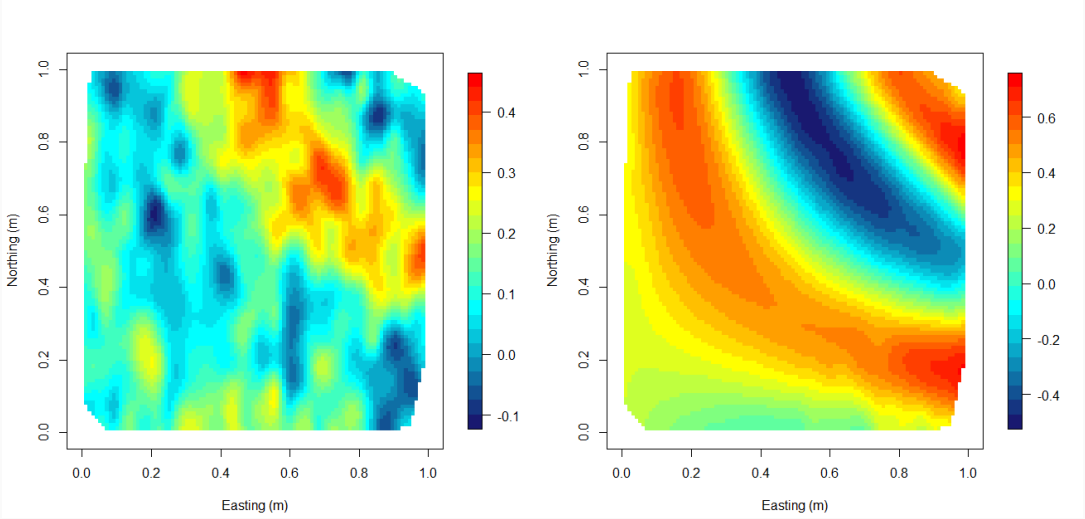
Figure: Pollutant levels in Europe in March, 2009

Point level modeling

- Point-level modeling refers to modeling of point-referenced data collected at locations referenced by coordinates (e.g., lat-long, Easting-Northing).
- Data from a spatial process $\{Y(\mathbf{s}) : \mathbf{s} \in \mathscr{D}\}$, $\mathscr{D}$ is a subset in Euclidean space.
- Example: $Y(\mathbf{s})$ is a pollutant level at site $\mathbf{s}$
- Conceptually: Pollutant level exists at all possible sites
- Practically: Data will be a partial realization of a spatial process – observed at $\{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$
- Statistical objectives: Inference about the process $Y(\mathbf{s})$; predict at new locations.
- Remarkable: Can learn about entire $Y(\mathbf{s})$ surface. The key: Structured dependence

Exploratory data analysis (EDA): Plotting the data

- A typical setup: Data observed at n locations $\{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$
- At each $\mathbf{s}_i$ we observe the response $y(\mathbf{s}_i)$ and a $p \times 1$ vector of covariates $\mathbf{x}(s_i)$
- Surface plots of the data often helps to understand spatial patterns



$y(\mathbf{s})$ $x(\mathbf{s})$

Figure: Response and covariate surface plots for Dataset 1

What's so special about spatial?

- Linear regression model: $y(\mathbf{s}_i) = \mathbf{x}(s_i)^\top \beta + \epsilon(\mathbf{s}_i)$
- $\epsilon(\mathbf{s}_i)$ are iid $N(0, \tau^2)$ errors
- $\mathbf{y} = (y(\mathbf{s}_1), \ldots, y(\mathbf{s}_n))^\top$; $\mathbf{X} = (\mathbf{x}(\mathbf{s}_1)^\top, \ldots, \mathbf{x}(\mathbf{s}_n)^\top)^\top$
- Inference: $\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \sim N(\beta, \tau^2 (\mathbf{X}^\top \mathbf{X})^{-1})$
- Prediction at new location $\mathbf{s}_0$: $\widehat{y(s_0)} = \mathbf{x}(s_0)^\top \hat{\beta}$
- Although the data is spatial, this is an ordinary linear regression model

Residual plots

- Surface plots of the residuals $(y(\mathbf{s}) - \widehat{y(\mathbf{s})})$ help to identify any spatial patterns left unexplained by the covariates

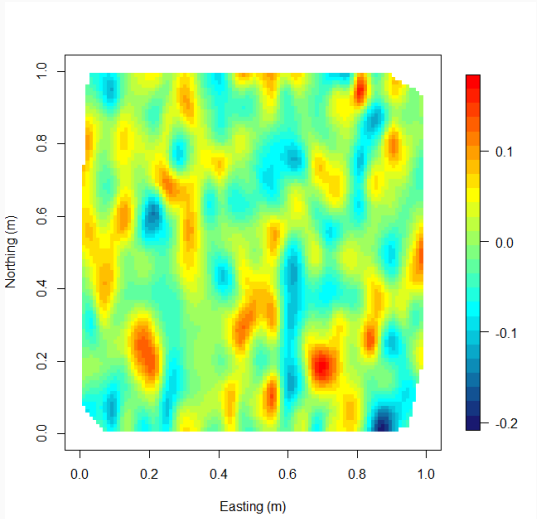


Figure: Residual plot for Dataset 1 after linear regression on $x(\mathbf{s})$

Residual plots

- Surface plots of the residuals $(y(\mathbf{s}) - \widehat{y(\mathbf{s})})$ help to identify any spatial patterns left unexplained by the covariates

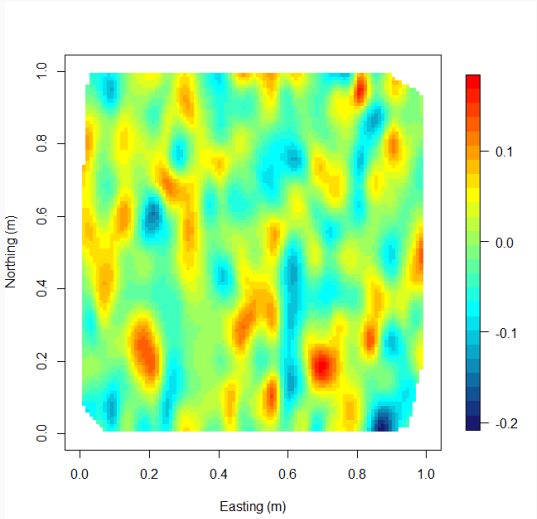
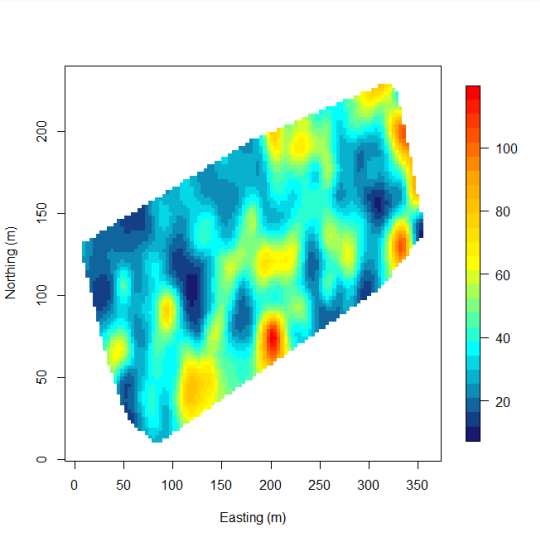


Figure: Residual plot for Dataset 1 after linear regression on $x(\mathbf{s})$

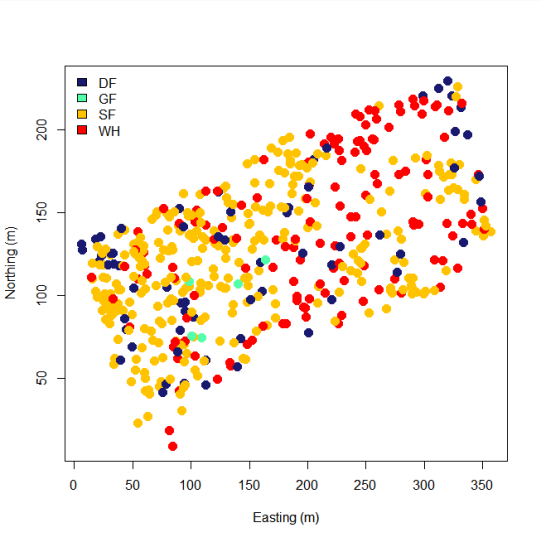
- No evident spatial pattern in plot of the residuals
- The covariate $x(\mathbf{s})$ seem to explain all spatial variation in $y(\mathbf{s})$
- Does a non-spatial regression model always suffice?

Western Experimental Forestry (WEF) data

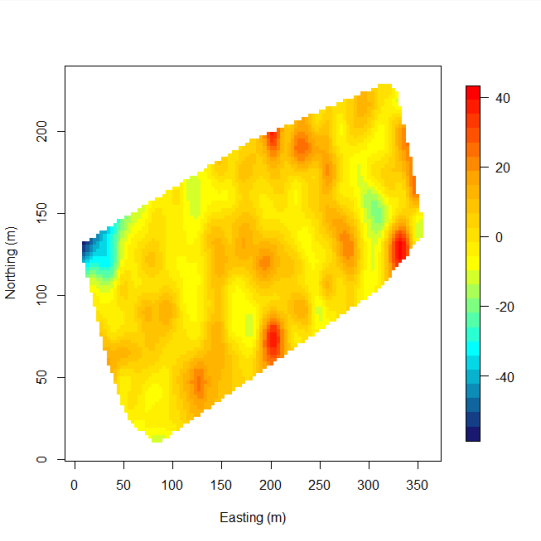
- Data consist of a census of all trees in a 10 ha. stand in Oregon
- Response of interest: Diameter at breast height (DBH)
- Covariate: Tree species (Categorical variable)



DBH



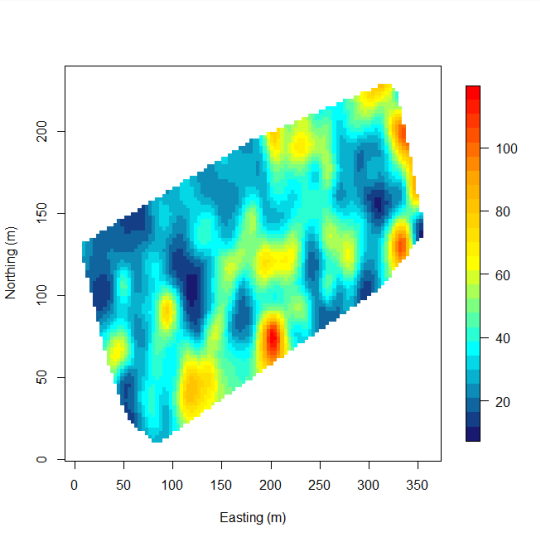
Species



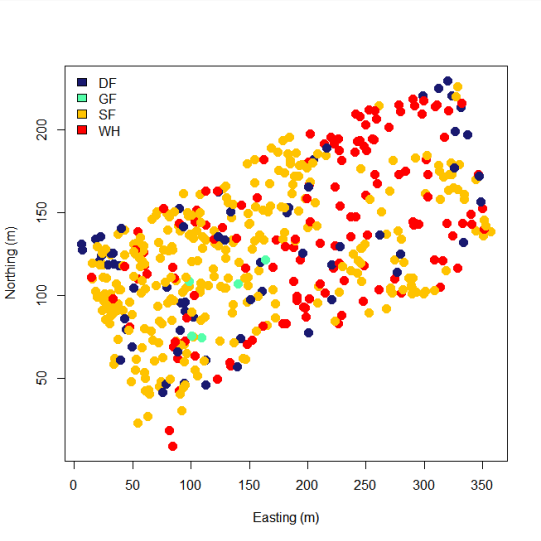
Residuals

Western Experimental Forestry (WEF) data

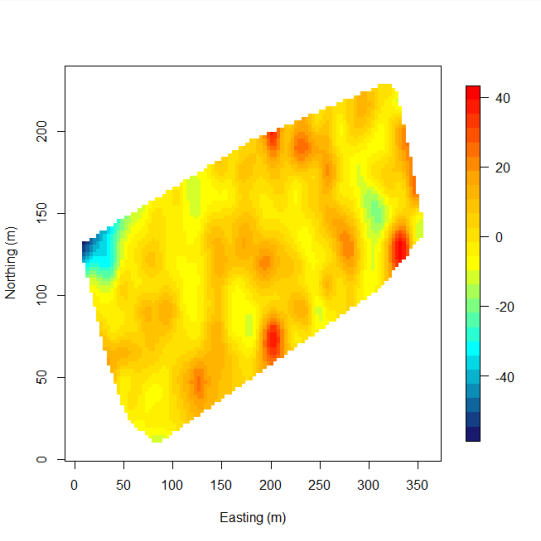
- Data consist of a census of all trees in a 10 ha. stand in Oregon
- Response of interest: Diameter at breast height (DBH)
- Covariate: Tree species (Categorical variable)



DBH



Species



Residuals

- Local spatial patterns in the residual plot
- Simple regression on species seems to be not sufficient

More EDA

- Besides eyeballing residual surfaces, how to do more formal EDA to identify spatial pattern?

More EDA

- Besides eyeballing residual surfaces, how to do more formal EDA to identify spatial pattern?

First law of geography

"Everything is related to everything else, but near things are more related than distant things." – Waldo Tobler

More EDA

- Besides eyeballing residual surfaces, how to do more formal EDA to identify spatial pattern?

First law of geography

"Everything is related to everything else, but near things are more related than distant things." – Waldo Tobler

- In general $(Y(\mathbf{s}+\mathbf{h}) - Y(\mathbf{s}))^2$ roughly increasing with $||\mathbf{h}||$ will imply a spatial correlation
- Can this be formalized to identify spatial pattern?

Empirical semivariogram

- Binning: Make intervals $I_1 = (0, m_1)$, $I_2 = (m_1, m_2)$, and so forth, up to $I_K = (m_{K-1}, m_K)$. Representing each interval by its midpoint t_k, we define:

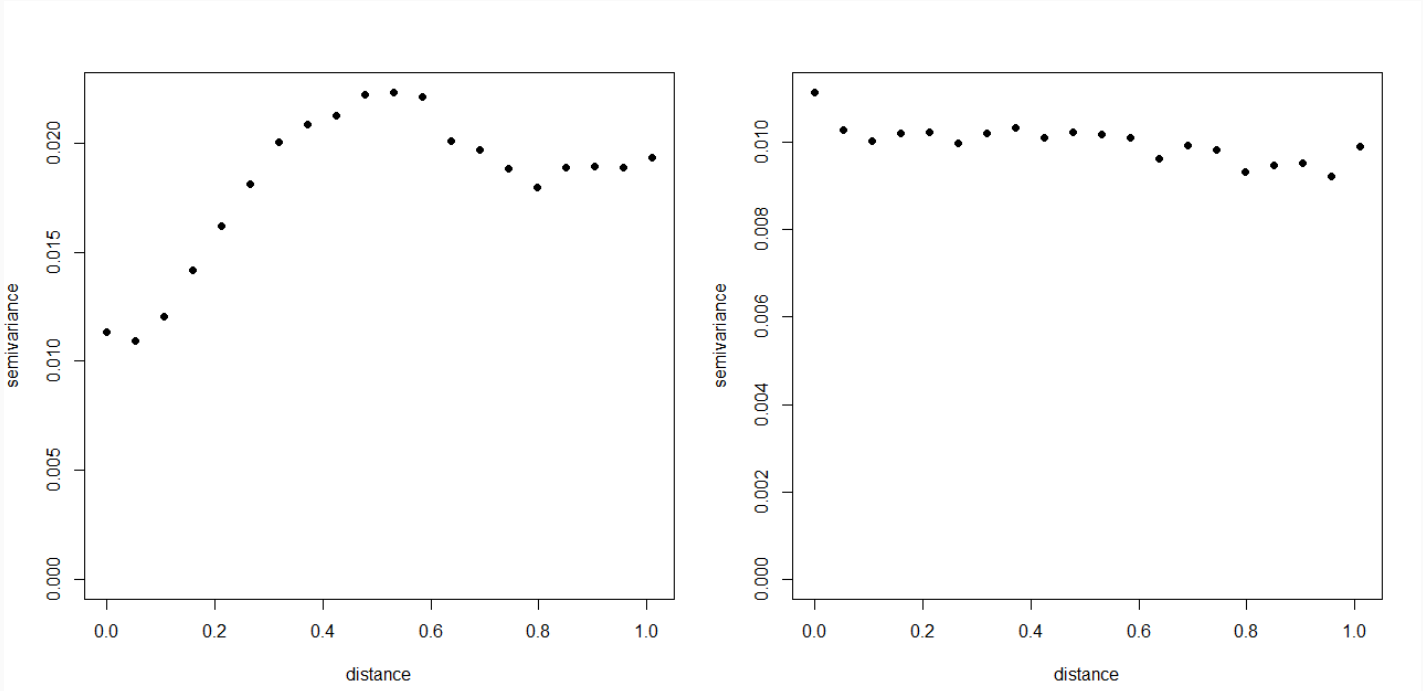
$$N(t_k) = \{(\mathbf{s}_i, \mathbf{s}_j) : \|\mathbf{s}_i - \mathbf{s}_j\| \in I_k\}, k = 1, \ldots, K.$$

- Empirical semivariogram:

$$\gamma(t_k) = \frac{1}{2|N(t_k)|} \sum_{\mathbf{s}_i, \mathbf{s}_j \in N(t_k)} (Y(\mathbf{s}_i) - Y(\mathbf{s}_j))^2$$

- For spatial data, the $\gamma(t_k)$ is expected to roughly increase with t_k
- A flat semivariogram would suggest little spatial variation

Empirical variogram: Data 1

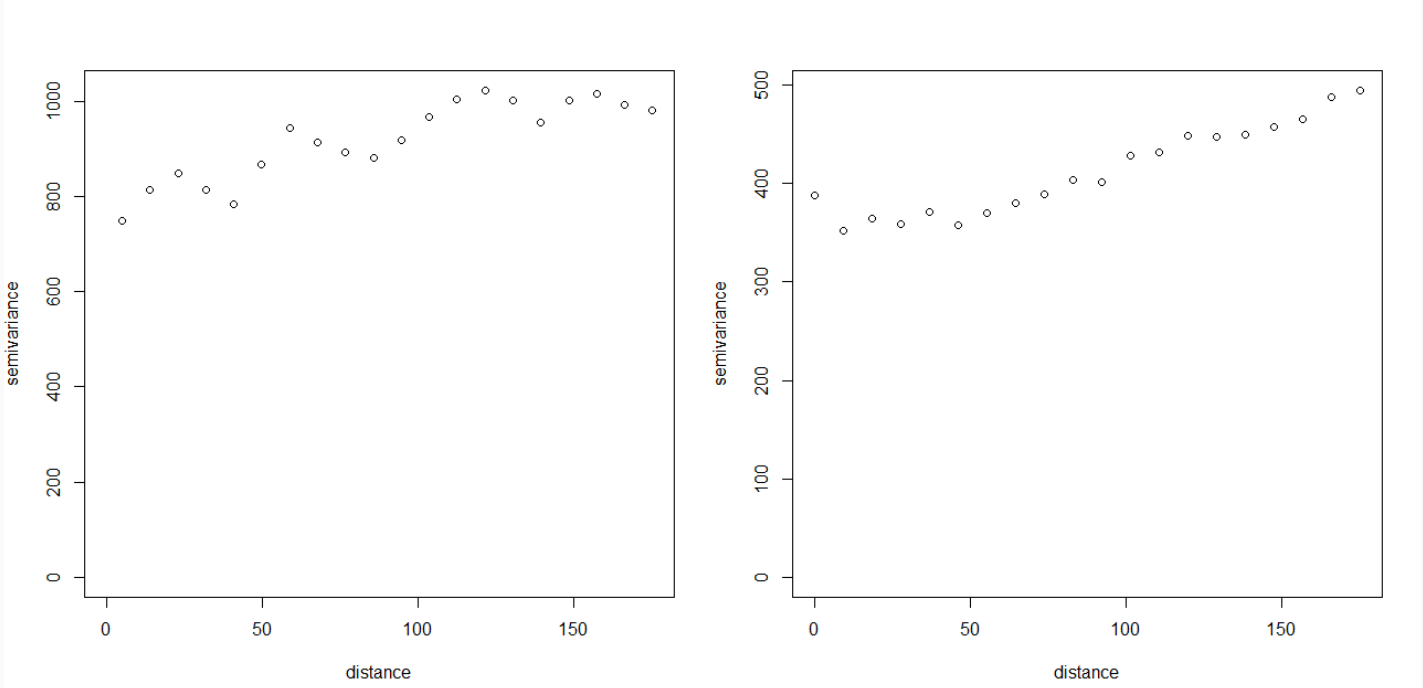


y residuals

- Residuals display little spatial variation

Empirical variograms: WEF data

- Regression model: DBH $\sim$ Species



DBH Residuals

- Variogram of the residuals confirm unexplained spatial variation

Modeling with the locations

- When purely covariate based models does not suffice, one needs to leverage the information from locations
- General model using the locations:
  $y(\mathbf{s}) = \mathbf{x}(\mathbf{s})^\top \boldsymbol{\beta} + w(\mathbf{s}) + \epsilon(\mathbf{s})$ for all $\mathbf{s} \in \mathscr{D}$
- How to choose the function $w(\cdot)$?
- Since we want to predict at any location over the entire domain $\mathscr{D}$, this choice will amount to choosing a surface $w(\mathbf{s})$
- How should such a surface be chosen?

Gaussian Processes (GPs)

- One popular approach to model $w(\mathbf{s})$ is via Gaussian Processes (GP)
- The collection of random variables $\{w(\mathbf{s}) \,|\, \mathbf{s} \in \mathscr{D}\}$ is a GP if
  - it is a valid stochastic process
  - all finite dimensional densities $\{w(\mathbf{s}_1), \ldots, w(\mathbf{s}_n)\}$ follow multivariate Gaussian distribution
- A GP is completely characterized by a mean function $m(\mathbf{s})$ and a covariance function $C(\cdot, \cdot)$
- Advantage: Likelihood based inference.
  $w = (\mathbf{w}(s_1), \ldots, w(\mathbf{s}_n))^\top \sim N(\mathbf{m}, \mathbf{C})$ where
  $\mathbf{m} = (m(\mathbf{s}_1), \ldots, m(\mathbf{s}_n))^\top$ and $\mathbf{C} = C(\mathbf{s}_i, \mathbf{s}_j)$

Valid covariance functions and isotropy

- $C(\cdot, \cdot)$ needs to be valid. For all n and all $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$, the resulting covariance matrix $C(\mathbf{s}_i, \mathbf{s}_j)$ for $(w(\mathbf{s}_1), w(\mathbf{s}_2), \ldots, w(\mathbf{s}_n))$ must be positive definite
- So, $C(\cdot, \cdot)$ needs to be a positive definite function
- Simplifying assumptions:
  - Stationarity: $C(\mathbf{s}_1, \mathbf{s}_2)$ only depends on $\mathbf{h} = \mathbf{s}_1 - \mathbf{s}_2$ (and is denoted by $C(\mathbf{h})$)
  - Isotropic: $C(\mathbf{h}) = C(||\mathbf{h}||)$
  - Anisotropic: Stationary but not isotropic
- Isotropic models are popular because of their simplicity, interpretability, and because a number of relatively simple parametric forms are available as candidates for C.

Some common isotropic covariance functions

Model	Covariance function, $C(t) = C(\|\|h\|\|)$
Spherical	$C(t) = \begin{cases} 0 & \text{if } t \geq 1/\phi \\ \sigma^2 \left[1 - \frac{3}{2}\phi t + \frac{1}{2}(\phi t)^3\right] & \text{if } 0 < t \leq 1/\phi \\ \tau^2 + \sigma^2 & \text{otherwise} \end{cases}$
Exponential	$C(t) = \begin{cases} \sigma^2 \exp(-\phi t) & \text{if } t > 0 \\ \tau^2 + \sigma^2 & \text{otherwise} \end{cases}$
Powered exponential	$C(t) = \begin{cases} \sigma^2 \exp(-\lvert\phi t\rvert^p) & \text{if } t > 0 \\ \tau^2 + \sigma^2 & \text{otherwise} \end{cases}$
Matérn at $\nu = 3/2$	$C(t) = \begin{cases} \sigma^2 (1 + \phi t) \exp(-\phi t) & \text{if } t > 0 \\ \tau^2 + \sigma^2 & \text{otherwise} \end{cases}$

Notes on exponential model

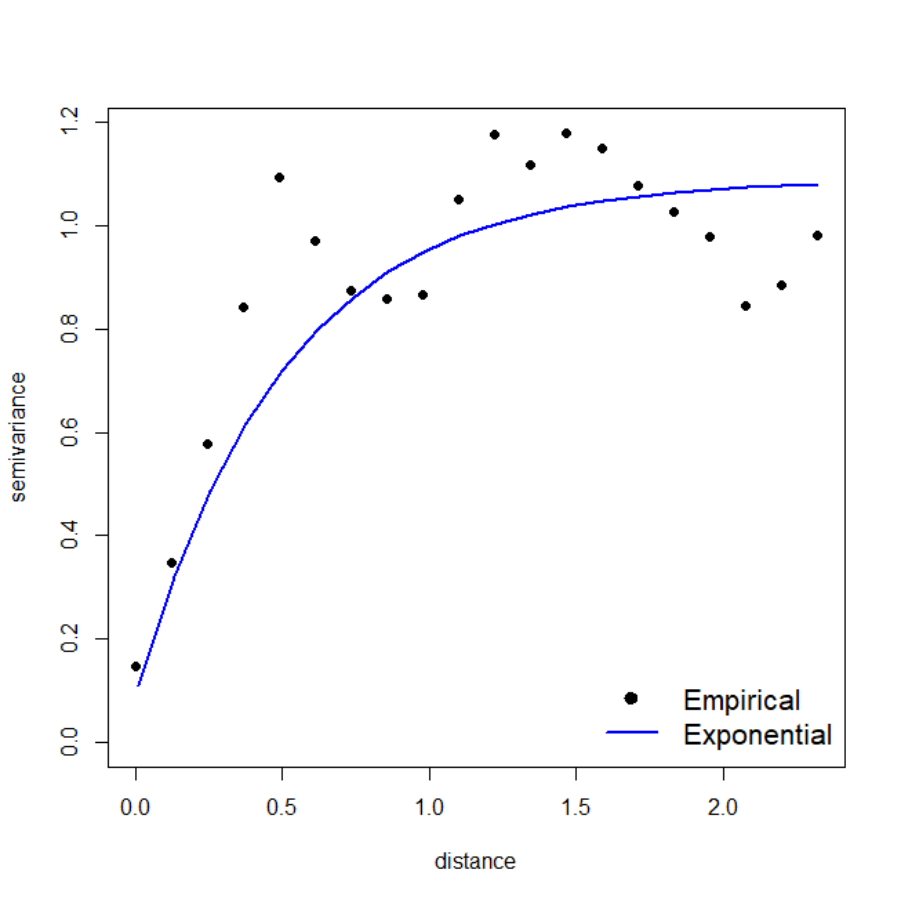
$$C(t) = \begin{cases} \tau^2 + \sigma^2 & \text{if } t = 0 \\ \sigma^2 \exp(-\phi t) & \text{if } t > 0 \end{cases}.$$

- We define the effective range, t_0, as the distance at which this correlation has dropped to only 0.05. Setting $\exp(-\phi t_0)$ equal to this value we obtain $t_0 \approx 3/\phi$, since $\log(0.05) \approx -3$.
- The nugget τ^2 is often viewed as a "nonspatial effect variance,"
- The partial sill (σ^2) is viewed as a "spatial effect variance."
- $\sigma^2 + \tau^2$ gives the maximum total variance often referred to as the sill
- Note discontinuity at 0 due to the nugget. Intentional! To account for measurement error or micro-scale variability.

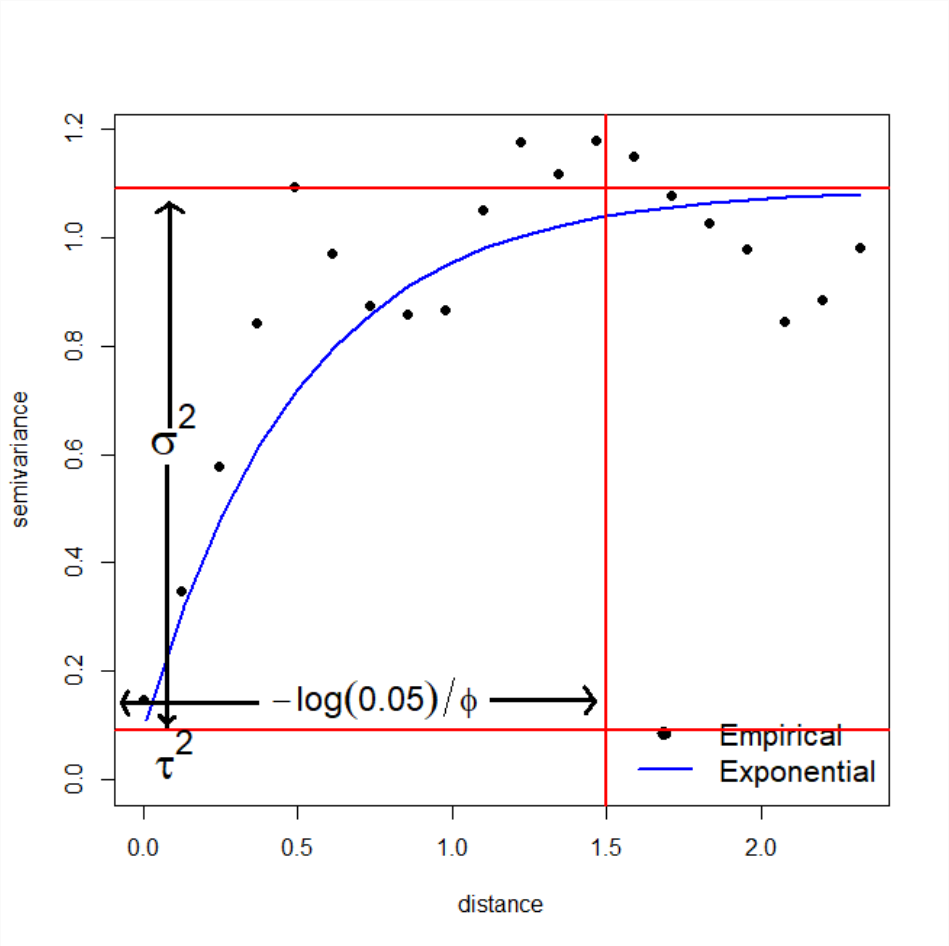
Covariance functions and semivariograms

- Recall: Empirical semivariogram:
  $\gamma(t_k) = \frac{1}{2|N(t_k)|} \sum_{\mathbf{s}_i, \mathbf{s}_j \in N(t_k)} (Y(\mathbf{s}_i) - Y(\mathbf{s}_j))^2$
- For any stationary GP,
  $E(Y(\mathbf{s}+\mathbf{h}) - Y(\mathbf{s}))^2/2 = C(\mathbf{0}) - C(\mathbf{h}) = \gamma(\mathbf{h})$
- $\gamma(\mathbf{h})$ is the semivariogram corresponding to the covariance function $C(\mathbf{h})$
- Example: For exponential GP,
  $$\gamma(t) = \begin{cases} \tau^2 + \sigma^2(1 - \exp(-\phi t)) & \text{if } t > 0 \\ 0 & \text{if } t = 0 \end{cases}, \text{ where } t = ||\mathbf{h}||$$

Covariance functions and semivariograms



Covariance functions and semivariograms



The Matèrn covariance function

- The Matèrn is a very versatile family:

$$C(t) = \begin{cases} \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)}(2\sqrt{\nu}t\phi)^\nu K_\nu(2\sqrt{(}\nu)t\phi) & \text{if } t > 0 \\ \tau^2 + \sigma^2 & \text{if } t = 0 \end{cases}$$

  K_ν is the modified Bessel function of order ν (computationally tractable)

- ν is a smoothness parameter controlling process smoothness. Remarkable!
- $\nu = 1/2$ gives the exponential covariance function

Kriging: Spatial prediction at new locations

- Goal: Given observations $\mathbf{w} = (w(\mathbf{s}_1), w(\mathbf{s}_2), \ldots, w(\mathbf{s}_n))^\top$, predict $w(\mathbf{s}_0)$ for a new location $\mathbf{s}_0$
- If $w(\mathbf{s})$ is modeled as a GP, then $(w(\mathbf{s}_0), w(\mathbf{s}_1), \ldots, w(\mathbf{s}_n))^\top$ jointly follow multivariate normal distribution
- $w(\mathbf{s}_0) \,|\, \mathbf{w}$ follows a normal distribution with
  - Mean (kriging estimator): $m(\mathbf{s}_0) + \mathbf{c}^\top \mathbf{C}^{-1}(\mathbf{w} - \mathbf{m})$
  - where $m = E(\mathbf{w})$, $\mathbf{C} = Cov(\mathbf{w})$, $\mathbf{c} = Cov(\mathbf{w}, w(\mathbf{s}_0))$
  - Variance: $\mathbf{C}(\mathbf{s}_0, \mathbf{s}_0) - \mathbf{c}^\top \mathbf{C}^{-1} \mathbf{c}$
- The GP formulation gives the full predictive distribution of $w(\mathbf{s}_0)|\mathbf{w}$

Modeling with GPs

Spatial linear model

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

- $w(\mathbf{s})$ modeled as $GP(0, C(\cdot \mid \boldsymbol{\theta}))$ (usually without a nugget)
- $\epsilon(\mathbf{s}) \overset{\text{iid}}{\sim} N(0, \tau^2)$ contributes to the nugget
- Under isotropy: $C(\mathbf{s} + \mathbf{h}, \mathbf{s}) = \sigma^2 R(||\mathbf{h}|| \,; \phi)$
- $\mathbf{w} = (w(\mathbf{s}_1), \ldots, w(\mathbf{s}_n))^\top \sim N(\mathbf{0}, \sigma^2 \mathbf{R}(\phi))$ where $\mathbf{R}(\phi) = \sigma^2(R(||s_i - s_j|| \,; \phi))$
- $\mathbf{y} = (y(\mathbf{s}_1), \ldots, y(\mathbf{s}_n))^\top \sim N(\mathbf{X}\beta, \sigma^2 \mathbf{R}(\phi) + \tau^2 \mathbf{I})$

Parameter estimation

- $\mathbf{y} = (y(\mathbf{s}_1), \ldots, y(\mathbf{s}_n))^\top \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{R}(\phi) + \tau^2\mathbf{I})$
- We can obtain MLEs of parameters $\boldsymbol{\beta}, \tau^2, \sigma^2, \phi$ based on the above model and use the estimates to krige at new locations
- In practice, the likelihood is often very flat with respect to the spatial covariance parameters and choice of initial values is important
- Initial values can be eyeballed from empirical semivariogram of the residuals from ordinary linear regression
- Estimated parameter values can be used for kriging

Model comparison

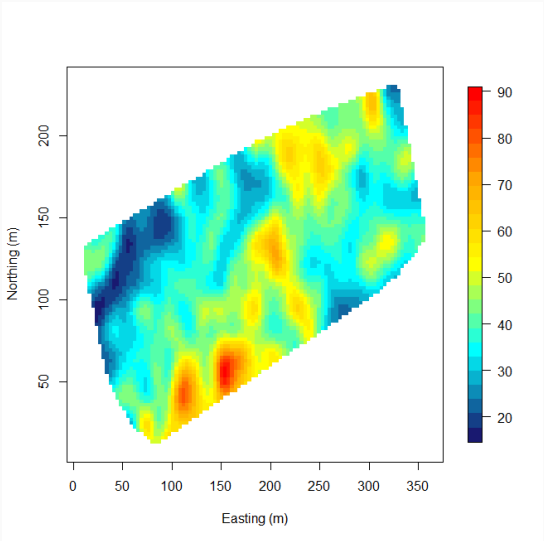
- For k total parameters and sample size n:
  - AIC: $2k - 2\log(l(\mathbf{y} \mid \hat{\beta}, \hat{\theta}, \hat{\tau^2}))$
  - BIC: $\log(n)k - 2\log(l(\mathbf{y} \mid \hat{\beta}, \hat{\theta}, \hat{\tau^2}))$
- Prediction based approaches using holdout data:
  - Root Mean Square Predictive Error (RMSPE): $\sqrt{\frac{1}{n_{out}} \sum_{i=1}^{n_{out}} (y_i - \hat{y}_i)^2}$
  - Coverage probability (CP): $\frac{1}{n_{out}} \sum_{i=1}^{n_{out}} I(y_i \in (\hat{y}_{i,0.025}, \hat{y}_{i,0.975}))$
  - Width of 95% confidence interval (CIW): $\frac{1}{n_{out}} \sum_{i=1}^{n_{out}} (\hat{y}_{i,0.975} - \hat{y}_{i,0.025})$
  - The last two approaches compares the distribution of y_i instead of comparing just their point predictions

Back to WEF data

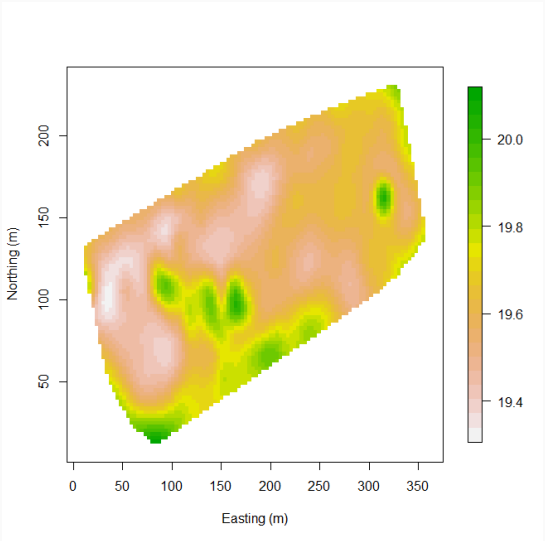
Table: Model comparison

	Spatial	Non-spatial
AIC	4419	4465
BIC	4448	4486
RMSPE	18	21
CP	93	93
CIW	77	82

WEF data: Kriged surfaces



DBH Estimates



Standard errors

Summary

- Geostatistics – Analysis of point-referenced spatial data
- Surface plots of data and residuals
- EDA with empirical semivariograms
- Modeling unknown surfaces with Gaussian Processes
- Kriging: Predictions at new locations
- Spatial linear regression using Gaussian Processes

Bayesian Linear Models

Andrew Finley[1] & Jeffrey Doser[2]

May 15, 2023

[1]Department of Forestry, Michigan State University.

[2]Department of Integrative Biology, Michigan State University.

Linear Regression

- Linear regression is, perhaps, *the* most widely used statistical modeling tool.
- It addresses the following question: How does a quantity of primary interest, y, vary as (depend upon) another quantity, or set of quantities, x?
- The quantity y is called the *response* or *outcome variable*. Some people simply refer to it as the *dependent variable*.
- The variable(s) x are called *explanatory variables*, *covariates* or simply *independent variables*.
- In general, we are interested in the conditional distribution of y, given x, parametrized as $p(y \mid \theta, x)$.

- Typically, we have a set of *units* or *experimental subjects* $i = 1, 2, \ldots, n$.

- For each of these units we have measured an outcome y_i and a set of explanatory variables $\mathbf{x}_i^\top = (1, x_{i1}, x_{i2}, \ldots, x_{ip})$.

- The first element of $\mathbf{x}_i^\top$ is often taken as 1 to signify the presence of an "intercept".

- We collect the outcome and explanatory variables into an $n \times 1$ vector and an $n \times (p+1)$ matrix:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}; \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1p} \\ 1 & x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{np} \end{bmatrix} = \begin{pmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix}.$$

- The linear model is the most fundamental of all serious statistical models underpinning:
    - ANOVA: y_i is continuous, x_{ij}'s are *all* categorical
    - REGRESSION: y_i is continuous, x_{ij}'s are continuous
    - ANCOVA: y_i is continuous, x_{ij}'s are continuous for some j and categorical for others.

Conjugate Bayesian Linear Regression

- A conjugate Bayesian linear model is given by:

$$y_i \,|\, \boldsymbol{\beta}, \sigma^2, \mathbf{x}_i \overset{ind}{\sim} N(\mu_i, \sigma^2); \quad i = 1, 2, \ldots, n \,;$$
$$\mu_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} = \mathbf{x}_i^\top \boldsymbol{\beta} \,; \quad \boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^\top \,;$$
$$\boldsymbol{\beta} \,|\, \sigma^2 \sim N(\boldsymbol{\mu}_\beta, \sigma^2 \mathbf{V}_\beta) \,; \quad \sigma^2 \sim IG(a, b) \,.$$

- Unknown parameters include the regression parameters and the variance, i.e. $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \sigma^2\}$.

- We assume $\mathbf{X}$ is observed without error and all inference is conditional on $\mathbf{X}$.

- The above model is often written in terms of the posterior density $p(\boldsymbol{\theta} \,|\, \mathbf{y}) \propto p(\boldsymbol{\theta}, \mathbf{y})$:

$$IG(\sigma^2 \,|\, a, b) \times N(\boldsymbol{\beta} \,|\, \boldsymbol{\mu}_\beta, \sigma^2 \mathbf{V}_\beta) \times \prod_{i=1}^{n} N(y_i \,|\, \mathbf{x}_i^\top \boldsymbol{\beta}, \sigma^2).$$

Conjugate Bayesian (General) Linear Regression

- A more general conjugate Bayesian linear model is given by:

$$\begin{aligned}
\mathbf{y} \,|\, \beta, \sigma^2, \mathbf{X} &\sim N(\mathbf{X}\beta, \sigma^2 \mathbf{V}_y) \\
\beta \,|\, \sigma^2 &\sim N(\boldsymbol{\mu}_\beta, \sigma^2 \mathbf{V}_\beta) \; ; \\
\sigma^2 &\sim IG(a, b) \; .
\end{aligned}$$

- $\mathbf{V}_y$, $\mathbf{V}_\beta$ and $\boldsymbol{\mu}_\beta$ are assumed fixed.
- Unknown parameters include the regression parameters and the variance, i.e. $\boldsymbol{\theta} = \{\beta, \sigma^2\}$.
- We assume $\mathbf{X}$ is observed without error and all inference is conditional on $\mathbf{X}$.
- The posterior density $p(\boldsymbol{\theta} \,|\, \mathbf{y}) \propto p(\boldsymbol{\theta}, \mathbf{y})$:

$$IG(\sigma^2 \,|\, a, b) \times N(\beta \,|\, \boldsymbol{\mu}_\beta, \sigma^2 \mathbf{V}_\beta) \times N(\mathbf{y} \,|\, \mathbf{X}\beta, \sigma^2 \mathbf{V}_y)$$

- The model on the previous slide is a special case with $\mathbf{V}_y = \mathbf{I}_n$ ($n \times n$ identity matrix).

Conjugate Bayesian (General) Linear Regression

- The joint posterior density can be written as

$$p(\beta, \sigma^2 \,|\, \mathbf{y}) \propto \underbrace{IG(\sigma^2 \,|\, a^*, b^*)}_{p(\sigma^2 \,|\, \mathbf{y})} \times \underbrace{N\left(\beta \,|\, \mathbf{Mm}, \sigma^2 \mathbf{M}\right)}_{p(\beta \,|\, \sigma^2, \mathbf{y})} ,$$

  where

$$a^* = a + \frac{n}{2} ; \quad b^* = b + \frac{1}{2}\left(\boldsymbol{\mu}_\beta^\top \mathbf{V}_\beta^{-1} \boldsymbol{\mu}_\beta + \mathbf{y}^\top \mathbf{V}_y^{-1} \mathbf{y} - \mathbf{m}^\top \mathbf{M} \mathbf{m}\right) ;$$

$$\mathbf{m} = \mathbf{V}_\beta^{-1} \boldsymbol{\mu}_\beta + \mathbf{X}^\top \mathbf{V}_y^{-1} \mathbf{y} ; \quad \mathbf{M}^{-1} = \mathbf{V}_\beta^{-1} + \mathbf{X}^\top \mathbf{V}_y^{-1} \mathbf{X} .$$

- Exact posterior sampling from $p(\beta, \sigma^2 \,|\, \mathbf{y})$ will automatically yield samples from $p(\beta \,|\, \mathbf{y})$ and $p(\sigma^2 \,|\, \mathbf{y})$.
- For each $j = 1, 2, \ldots, N$ do the following:
  1. Draw $\sigma^2_{(j)} \sim IG(a^*, b^*)$
  2. Draw $\beta_{(j)} \sim N\left(\mathbf{Mm}, \sigma^2_{(j)} \mathbf{M}\right)$
- The above is sometimes referred to as *composition sampling*.

Exact sampling from joint posterior distributions

- Suppose we wish to draw samples from a joint posterior:

$$p(\theta_1, \theta_2 \,|\, \mathbf{y}) = p(\theta_1 \,|\, \mathbf{y}) \times p(\theta_2 \,|\, \theta_1, \mathbf{y}) \; .$$

- In conjugate models, it is often easy to draw samples from $p(\theta_1 \,|\, \mathbf{y})$ and from $p(\theta_2 \,|\, \theta_1, \mathbf{y})$.
- We can draw N samples from $p(\theta_1, \theta_2 \,|\, \mathbf{y})$ as follows.
- For each $j = 1, 2, \ldots, N$ do the following:
  1. Draw $\theta_{1(j)} \sim p(\theta_1 \,|\, \mathbf{y})$
  2. Draw $\theta_{2(j)} \sim p(\theta_2 \,|\, \theta_{1(j)}, \mathbf{y})$
- Remarkably, the $\theta_{2(j)}$'s drawn above have marginal distribution $p(\theta_2 \,|\, \mathbf{y})$ (see, Gelfand and Smith 1990).
- "Automatic Marginalization" we draw samples $p(\theta_1, \theta_2 \,|\, \mathbf{y})$ and automatically get samples from $p(\theta_1 \,|\, \mathbf{y})$ and $p(\theta_2 \,|\, \mathbf{y})$.

Bayesian predictions from linear regression

- Let $\tilde{\mathbf{y}}$ denote an $m \times 1$ vector of outcomes we seek to predict based upon predictors $\tilde{\mathbf{X}}$.
- We seek the posterior predictive density:

$$p(\tilde{\mathbf{y}} \,|\, \mathbf{y}) = \int p(\tilde{\mathbf{y}} \,|\, \boldsymbol{\theta}, \mathbf{y}) p(\boldsymbol{\theta} \,|\, \mathbf{y}) \mathrm{d}\boldsymbol{\theta}.$$

- Posterior predictive inference: sample from $p(\tilde{\mathbf{y}} \,|\, \mathbf{y})$.
- For each $j = 1, 2, \ldots, N$ do the following:
  1. Draw $\boldsymbol{\theta}_{(j)} \sim p(\boldsymbol{\theta} \,|\, \mathbf{y})$
  2. Draw $\tilde{\mathbf{y}}_{(j)} \sim p(\tilde{\mathbf{y}} \,|\, \boldsymbol{\theta}_{(j)}, \mathbf{y})$

Bayesian predictions from linear regression (cont'd)

- For legitimate probabilistic predictions (forecasting), the conditional distribution $p(\tilde{\mathbf{y}} \,|\, \boldsymbol{\theta}, \mathbf{y})$ must be well-defined.
- For example, consider the case with $\mathbf{V}_y = \mathbf{I}_n$. Specify the linear model:

$$\begin{bmatrix} \mathbf{y} \\ \tilde{\mathbf{y}} \end{bmatrix} = \begin{bmatrix} \mathbf{X} \\ \tilde{\mathbf{X}} \end{bmatrix} \beta + \begin{bmatrix} \boldsymbol{\epsilon} \\ \tilde{\boldsymbol{\epsilon}} \end{bmatrix} \; ; \quad \begin{bmatrix} \boldsymbol{\epsilon} \\ \tilde{\boldsymbol{\epsilon}} \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \sigma^2 \begin{bmatrix} \mathbf{I}_n & O \\ O & \mathbf{I}_m \end{bmatrix} \right) .$$

- Easy to derive the conditional density:

$$p(\tilde{\mathbf{y}} \,|\, \boldsymbol{\theta}, \mathbf{y}) = p(\tilde{\mathbf{y}} \,|\, \boldsymbol{\theta}) = N(\tilde{\mathbf{y}} \,|\, \tilde{\mathbf{X}}\beta, \sigma^2 \mathbf{I}_m)$$

- Posterior predictive density:

$$p(\tilde{\mathbf{y}} \,|\, \mathbf{y}) = \int N(\tilde{\mathbf{y}} \,|\, \tilde{\mathbf{X}}\beta, \sigma^2 \mathbf{I}_m) p(\beta, \sigma^2 \,|\, \mathbf{y}) \mathrm{d}\beta \mathrm{d}\sigma^2 \; .$$

- For each $j = 1, 2, \ldots, N$ do the following:
  1. Draw $\{\beta_{(j)}, \sigma^2_{(j)}\} \sim p(\beta, \sigma^2 \,|\, \mathbf{y})$
  2. Draw $\tilde{\mathbf{y}}_{(j)} \sim N(\tilde{\mathbf{X}}\beta_{(j)}, \sigma^2_{(j)} \mathbf{I}_m)$

Bayesian predictions from general linear regression

- For example, consider the case with general $\mathbf{V}_y$. Specify:

$$\begin{bmatrix}\mathbf{y}\\ \tilde{\mathbf{y}}\end{bmatrix} = \begin{bmatrix}\mathbf{X}\\ \tilde{\mathbf{X}}\end{bmatrix}\beta + \begin{bmatrix}\boldsymbol{\epsilon}\\ \tilde{\boldsymbol{\epsilon}}\end{bmatrix}\ ; \quad \begin{bmatrix}\boldsymbol{\epsilon}\\ \tilde{\boldsymbol{\epsilon}}\end{bmatrix} \sim N\left(\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \sigma^2\begin{bmatrix}\mathbf{V}_y & \mathbf{V}_{y\tilde{y}}\\ \mathbf{V}_{y\tilde{y}}^\top & \mathbf{V}_{\tilde{y}}\end{bmatrix}\right)\ .$$

- Derive the conditional density $p(\tilde{\mathbf{y}} \mid \boldsymbol{\theta}, \mathbf{y}) = N\left(\tilde{\mathbf{y}} \mid \boldsymbol{\mu}_{\tilde{y}|y}, \sigma^2\mathbf{V}_{\tilde{y}|y}\right)$:

$$\boldsymbol{\mu}_{\tilde{y}|y} = \tilde{\mathbf{X}}\beta + \mathbf{V}_{y\tilde{y}}^\top\mathbf{V}_y^{-1}(\mathbf{y} - \mathbf{X}\beta)\ ; \quad \mathbf{V}_{\tilde{y}|y} = \mathbf{V}_{\tilde{y}} - \mathbf{V}_{y\tilde{y}}^\top\mathbf{V}_y^{-1}\mathbf{V}_{y\tilde{y}}.$$

- Posterior predictive density:

$$p(\tilde{\mathbf{y}} \mid \mathbf{y}) = \int N\left(\tilde{\mathbf{y}} \mid \boldsymbol{\mu}_{\tilde{y}|y}, \sigma^2\mathbf{V}_{\tilde{y}|y}\right) p(\beta, \sigma^2 \mid \mathbf{y})\mathrm{d}\beta\mathrm{d}\sigma^2.$$

- For each $j = 1, 2, \ldots, N$ do the following:
  1. Draw $\{\beta_{(j)}, \sigma^2_{(j)}\} \sim p(\beta, \sigma^2 \mid \mathbf{y})$
  2. Compute $\boldsymbol{\mu}_{\tilde{y}|y}$ using $\beta_{(j)}$ and draw $\tilde{\mathbf{y}}_{(j)} \sim N(\boldsymbol{\mu}_{\tilde{y}|y}, \sigma^2_{(j)}\mathbf{V}_{\tilde{y}})$

Application to Bayesian Geostatistics

- Consider the spatial regression model

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

  where $w(\mathbf{s}_i)$'s are spatial random effects and $\epsilon(\mathbf{s}_i)$'s are unstructured errors ("white noise").
- $\mathbf{w} = (w(\mathbf{s}_1), w(\mathbf{s}_2), \ldots, w(\mathbf{s}_n))^\top \sim N(\mathbf{0}, \sigma^2\mathbf{R}(\phi))$
- $\boldsymbol{\epsilon} = (\epsilon(\mathbf{s}_1), \epsilon(\mathbf{s}_2), \ldots, \epsilon(\mathbf{s}_n))^\top \sim N(\mathbf{0}, \tau^2\mathbf{I}_n)$
- Integrating out random effects leads to a Bayesian model:

$$IG(\sigma^2 \mid a, b) \times N(\boldsymbol{\beta} \mid \boldsymbol{\mu}_\beta, \sigma^2\mathbf{V}_\beta) \times N(\mathbf{y} \mid \mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{V}_y)$$

  where $\mathbf{V}_y = \mathbf{R}(\phi) + \alpha\mathbf{I}_n$ and $\alpha = \tau^2/\sigma^2$.
- Fixing ϕ and α (e.g., from variogram or other EDA) yields a conjugate Bayesian model (see `bayesGeostatExact()` in `spBayes` package).
- Exact posterior sampling is easily achieved as before!

Inference on spatial random effects

- Rewrite the model in terms of $\mathbf{w}$ as:

$$IG(\sigma^2 \mid a, b) \times N(\beta \mid \boldsymbol{\mu}_\beta, \sigma^2 \mathbf{V}_\beta) \times N(\mathbf{w} \mid \mathbf{0}, \sigma^2 \mathbf{R}(\phi)) \times N(\mathbf{y} \mid \mathbf{X}\beta + \mathbf{w}, \tau^2 \mathbf{I}_n).$$

- Posterior distribution of spatial random effects $\mathbf{w}$:

$$p(\mathbf{w} \mid \mathbf{y}) = \int N(\mathbf{w} \mid \mathbf{M}\mathbf{m}, \sigma^2 \mathbf{M}) \times p(\beta, \sigma^2 \mid \mathbf{y}) \mathrm{d}\beta \mathrm{d}\sigma^2 \ ,$$

  where $\mathbf{m} = (1/\alpha)(\mathbf{y} - \mathbf{X}\beta)$ and $\mathbf{M}^{-1} = \mathbf{R}^{-1}(\phi) + (1/\alpha)\mathbf{I}_n$.

- For each $j = 1, 2, \ldots, N$ do the following:
  1. Draw $\{\beta_{(j)}, \sigma^2_{(j)}\} \sim p(\beta, \sigma^2 \mid \mathbf{y})$
  2. Compute $\mathbf{m}$ from $\beta_{(j)}$ and draw $\mathbf{w}_{(j)} \sim N(\mathbf{M}\mathbf{m}, \sigma^2_{(j)}\mathbf{M})$

Inference on the process

- Posterior distribution of $w(\mathbf{s}_0)$ at new location $\mathbf{s}_0$:

$$p(w(\mathbf{s}_0) \mid \mathbf{y}) = \int N(w(\mathbf{s}_0) \mid \mu_{w(\mathbf{s}_0)|w}, \sigma^2_{w(\mathbf{s}_0)|w}) \times p(\sigma^2, \mathbf{w} \mid \mathbf{y}) \mathrm{d}\sigma^2 \mathrm{d}\mathbf{w} \ ,$$

  where

$$\begin{aligned} \mu_{w(\mathbf{s}_0)|w} &= \mathbf{r}^\top(\mathbf{s}_0; \phi)\mathbf{R}^{-1}(\phi)\mathbf{w} \ ; \\ \sigma^2_{w(\mathbf{s}_0)|w} &= \sigma^2\{1 - \mathbf{r}^\top(\mathbf{s}_0; \phi)\mathbf{R}^{-1}(\phi)\mathbf{r}(\mathbf{s}_0, \phi)\} \end{aligned}$$

- For each $j = 1, 2, \ldots, N$ do the following:
  1. Compute $\mu_{w(\mathbf{s}_0)|w}$ and $\sigma^2_{w(\mathbf{s}_0)|w}$ from $\mathbf{w}_{(j)}$ and $\sigma^2_{(j)}$.
  2. Draw $w_{(j)}(\mathbf{s}_0) \sim N(\mu_{w(\mathbf{s}_0)|w}, \sigma^2_{w(\mathbf{s}_0)|w})$.

Bayesian "kriging" or prediction

- Posterior predictive distribution at new location $\mathbf{s}_0$ is $p(y(\mathbf{s}_0) \mid \mathbf{y})$:

$$\int N(y(\mathbf{s}_0) \mid \mathbf{x}^\top(s_0)\beta + w(\mathbf{s}_0), \alpha\sigma^2) \times p(\beta, \sigma^2, \mathbf{w} \mid \mathbf{y}) \mathrm{d}\beta \mathrm{d}\sigma^2 \mathrm{d}\mathbf{w} \ ,$$

- For each $j = 1, 2, \ldots, N$ do the following:
  1. Draw $y_{(j)}(\mathbf{s}_0) \sim N(\mathbf{x}^\top(\mathbf{s}_0)\beta_{(j)} + w_{(j)}(s_0), \alpha\sigma^2_{(j)})$.

Non-conjugate models: The Gibbs Sampler

- Let $\theta = (\theta_1, \ldots, \theta_p)$ be the parameters in our model.
- Initialize with starting values $\theta^{(0)} = (\theta_1^{(0)}, \ldots, \theta_p^{(0)})$
- For $j = 1, \ldots, N$, update successively using the *full conditional* distributions:

$$\theta_1^{(j)} \sim p(\theta_1^{(j)} \mid \theta_2^{(j-1)}, \ldots, \theta_p^{(j-1)}, \mathbf{y})$$
$$\theta_2^{(j)} \sim p(\theta_2 \mid \theta_1^{(j)}, \theta_3^{(j-1)}, \ldots, \theta_p^{(j-1)}, \mathbf{y})$$
$$\vdots$$
(the generic k^{th} element)
$$\theta_k^{(j)} \sim p(\theta_k | \theta_1^{(j)}, \ldots, \theta_{k-1}^{(j)}, \theta_{k+1}^{(j-1)}, \ldots, \theta_p^{(j-1)}, \mathbf{y})$$
$$\vdots$$
$$\theta_p^{(j)} \sim p(\theta_p \mid \theta_1^{(j)}, \ldots, \theta_{p-1}^{(j)}, \mathbf{y})$$

- In principle, the Gibbs sampler will work for extremely complex hierarchical models. The only issue is sampling from the full conditionals. They may not be amenable to easy sampling – when these are not in closed form. A more general and extremely powerful - and often easier to code - algorithm is the Metropolis-Hastings (MH) algorithm.

- This algorithm also constructs a Markov chain, but does not necessarily care about full conditionals.

- Popular approach: Embed Metropolis steps within Gibbs to draw from full conditionals that are not accessible to directly generate from.

When we don't want to fix ϕ and $\alpha = \tau^2/\sigma^2$

Latent Bayesian Model

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

Sampler:

- Full conditionals for β, τ^2, σ^2 and $w(\mathbf{s}_i)$'s
- Metropolis step for updating ϕ
- Pros: Full conditional distributions for all parameters except ϕ, easy to code up
- Cons: High-dimensional parameter space can mean slow convergence

When we don't want to fix ϕ and $\alpha = \tau^2/\sigma^2$ (cont'd)

Collapsed Bayesian Model

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

Sampler:

- Full conditional for β
- Metropolis step for updating τ^2, σ^2, ϕ
- Pros: Low-dimensional parameter space
- "Recover" $w(\mathbf{s}_i)$'s in a posterior predictive fashion

We can also integrate out β! See Finley et al. (2015) for details https://www.jstatsoft.org/article/view/v063i13 and implementation in the `spBayes` package.

The Metropolis-Hastings Algorithm

- The Metropolis-Hastings algorithm: Start with a initial value for $\theta = \theta^{(0)}$. Select a *candidate* or *proposal* distribution from which to propose a value of θ at the j-th iteration: $\theta^{(j)} \sim q(\theta^{(j-1)}, \nu)$. For example, $q(\theta^{(j-1)}, \nu) = N(\theta^{(j-1)}, \nu)$ with ν fixed.
- Compute
$$r = \frac{p(\theta^* \mid y) q(\theta^{(j-1)} \mid \theta^*, \nu)}{p(\theta^{(j-1)} \mid y) q(\theta^* \mid \theta^{(j-1)} \nu)}$$
- If $r \geq 1$ then set $\theta^{(j)} = \theta^*$. If $r \leq 1$ then draw $U \sim (0, 1)$. If $U \leq r$ then $\theta^{(j)} = \theta^*$. Otherwise, $\theta^{(j)} = \theta^{(j-1)}$.
- Repeat for $j = 1, \ldots N$. This yields $\theta^{(1)}, \ldots, \theta^{(N)}$, which, after a burn-in period, will be samples from the true posterior distribution. It is important to monitor the acceptance ratio r of the sampler through the iterations. Rough recommendations: for vector updates $r \approx 20\%$., for scalar updates $r \approx 40\%$. This can be controlled by "tuning" ν.
- Popular approach: Embed Metropolis steps within Gibbs to draw from full conditionals that are not accessible to directly generate from.

- Example: For the linear model, our parameters are (β, σ^2). We write $\theta = (\beta, \log(\sigma^2))$ and, at the j-th iteration, propose $\theta^* \sim N(\theta^{(j-1)}, \Sigma)$. The log transformation on σ^2 ensures that all components of θ have support on the entire real line and can have meaningful proposed values from the multivariate normal. But we need to transform our prior to $p(\beta, \log(\sigma^2))$.
- Let $z = \log(\sigma^2)$ and assume $p(\beta, z) = p(\beta)p(z)$. Let us derive $p(z)$. REMEMBER: we need to adjust for the jacobian. Then $p(z) = p(\sigma^2)|d\sigma^2/dz| = p(e^z)e^z$. The jacobian here is $e^z = \sigma^2$.
- Let $p(\beta) = 1$ and an $p(\sigma^2) = IG(\sigma^2 \mid a, b)$. Then log-posterior is:

$$-(a + n/2 + 1)z + z - \frac{1}{e^z}\{b + \frac{1}{2}(Y - X\beta)^T(Y - X\beta)\}.$$

- A symmetric proposal distribution, say $q(\theta^*|\theta^{(j-1)}, \Sigma) = N(\theta^{(j-1)}, \Sigma)$, cancels out in r. In practice it is better to compute $\log(r)$: $\log(r) = \log(p(\theta^* \mid y) - \log(p(\theta^{(j-1)} \mid y))$. For the proposal, $N(\theta^{(j-1)}, \Sigma)$, Σ is a $d \times d$ variance-covariance matrix, and $d = \dim(\theta) = p + 1$.
- If $\log r \geq 0$ then set $\theta^{(j)} = \theta^*$. If $\log r \leq 0$ then draw $U \sim (0, 1)$. If $U \leq r$ (or $\log U \leq \log r$) then $\theta^{(j)} = \theta^*$. Otherwise, $\theta^{(j)} = \theta^{(j-1)}$.
- Repeat the above procedure for $j = 1, \ldots N$ to obtain samples $\theta^{(1)}, \ldots, \theta^{(N)}$.

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

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

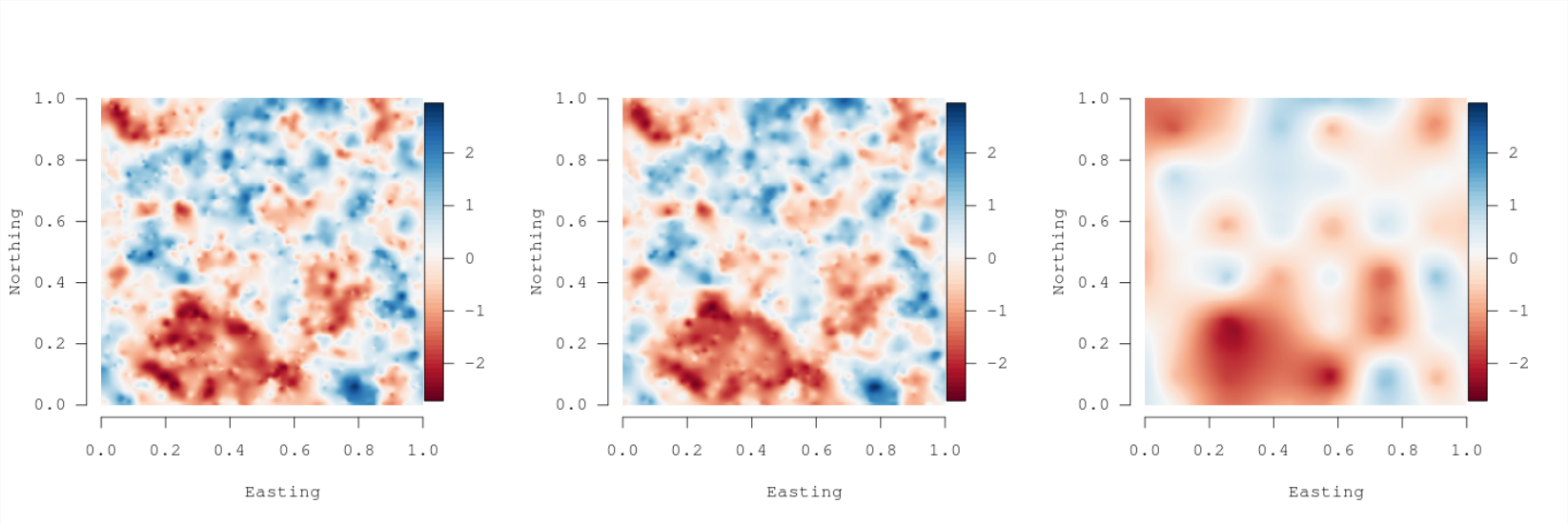
  This is the famous Sherman-Woodbury-Morrison formula.

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.

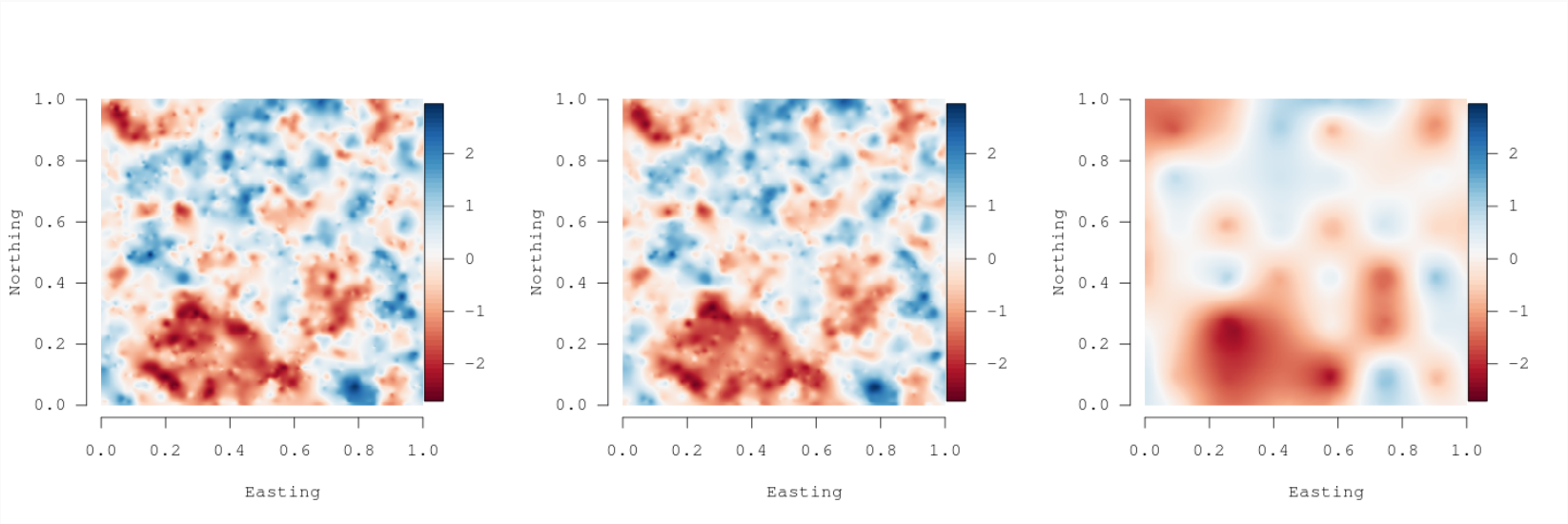
Low rank Gaussian Predictive Process

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

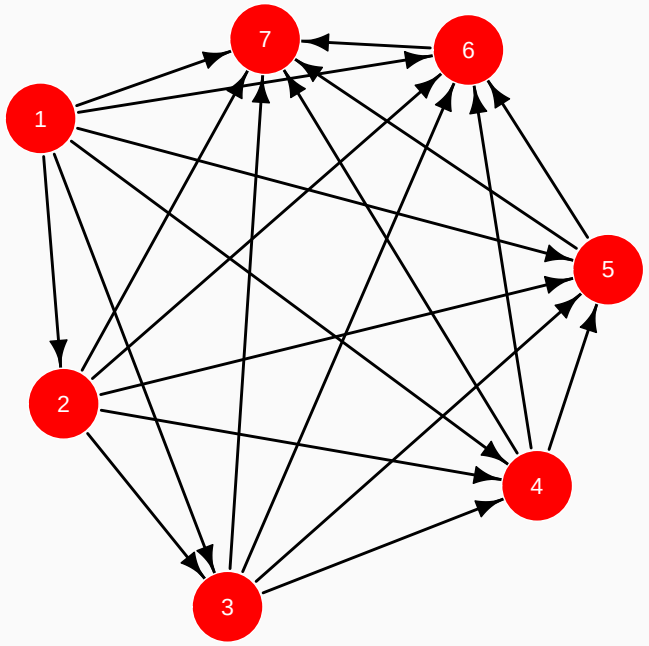
$$\sum_{j<i} a_{ij} w_j = \mathrm{E}[w_i \mid w_{\{j<i\}}] \quad i = 2, \ldots, n$$

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

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

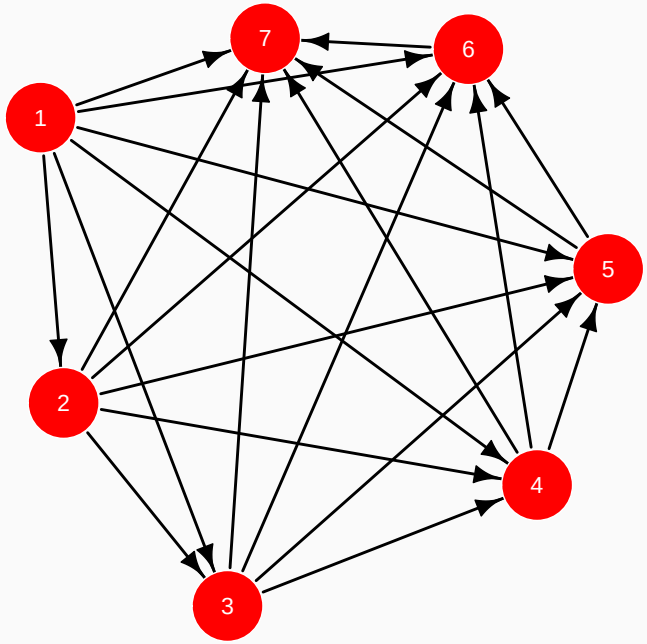
```
for(i in 1:(n-1)) {
  A[i+1,1:i] = solve(K[1:i,1:i], K[1:i,i+1])
  D[i+1,i+1] = K[i+1,i+1] - dot(K[i+1,1:i],A[i+1,1:i])
}
```

Cholesky Factors and Directed Acyclic Graphs (DAGs)



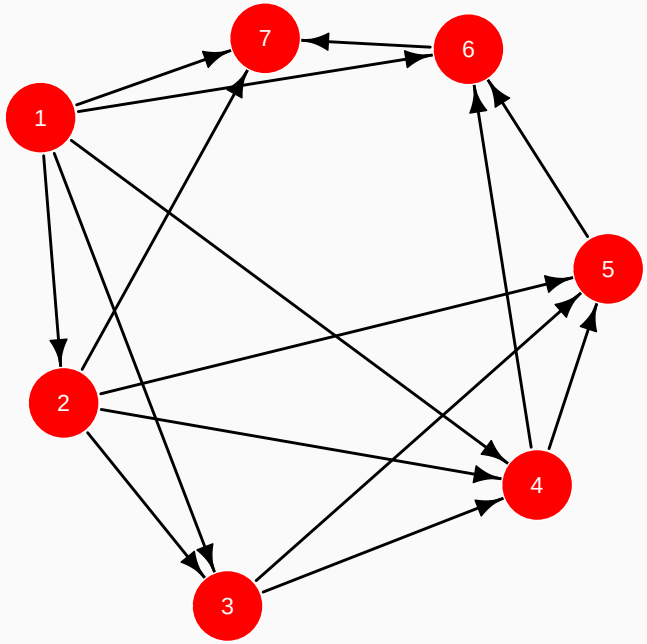
- Number of non-zero entries (sparsity) of **A** equals number of arrows in the graph
- In particular: Sparsity of the i^{th} row of **A** is same as the number of arrows towards i in the DAG

Introducing sparsity via graphical models



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

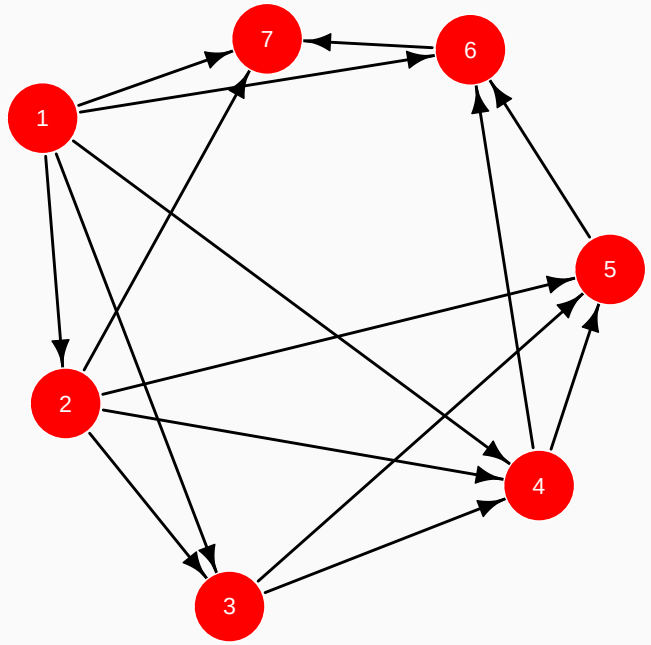
Introducing sparsity via graphical models



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

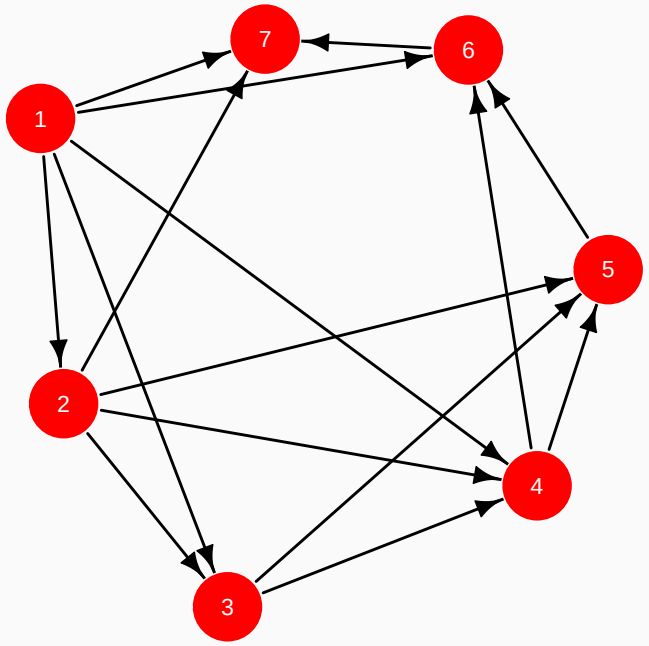
$$p(y_5 \mid \cancel{y_1}, y_2, y_3, y_4)p(y_6 \mid y_1, \cancel{y_2}, \cancel{y_3}, y_4, y_5)p(y_7 \mid y_1, y_2, \cancel{y_3}, \cancel{y_4}, \cancel{y_5}, y_6)$$

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:

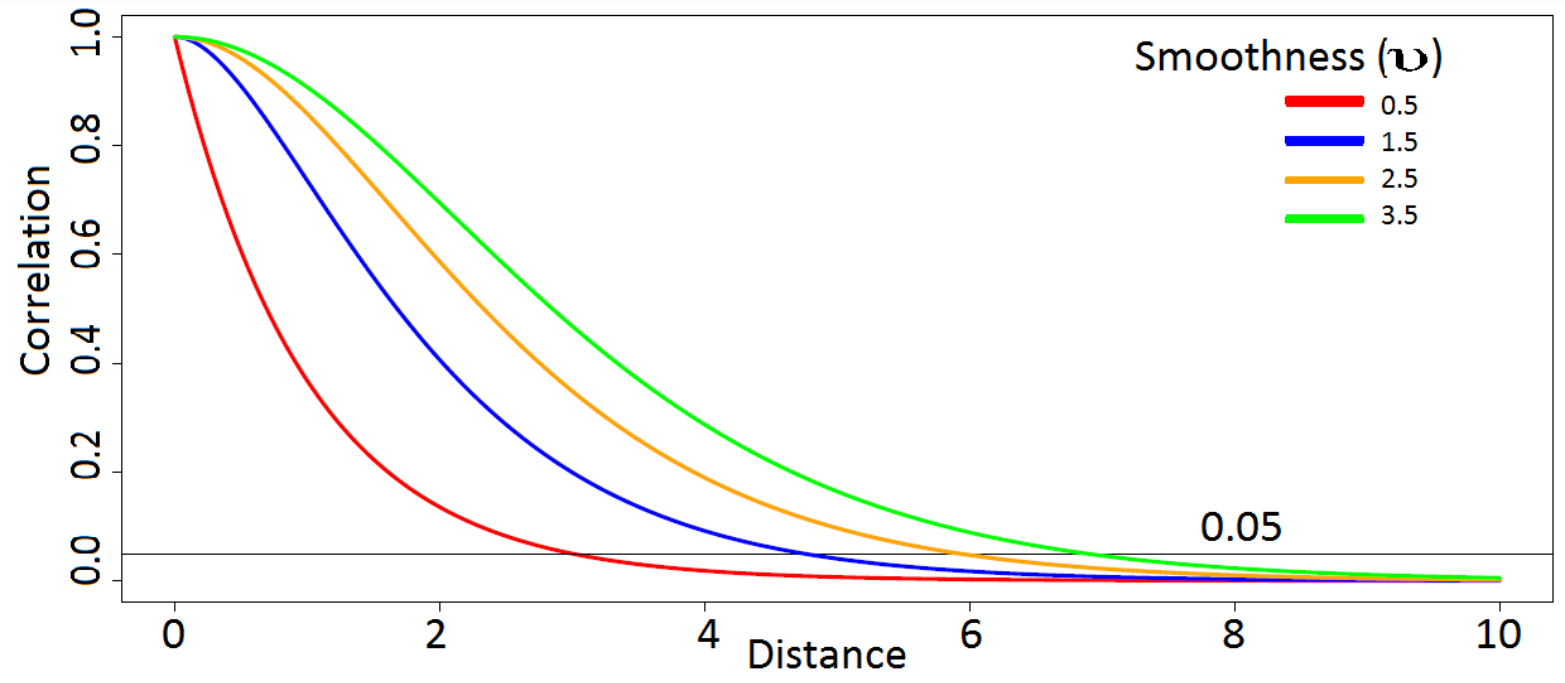
```
D[1,1] = K[1,1] and first row of A is zero.
for(i in 1:(n-1)) {
  Pa = N[i+1] # neighbors of i+1
  A[i+1,Pa] = solve(K[Pa,Pa], K[i+1,Pa])
  D[i+1,i+1] = K[i+1,i+1] - dot(K[i+1,Pa],A[i+1,Pa])
}
```

- 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

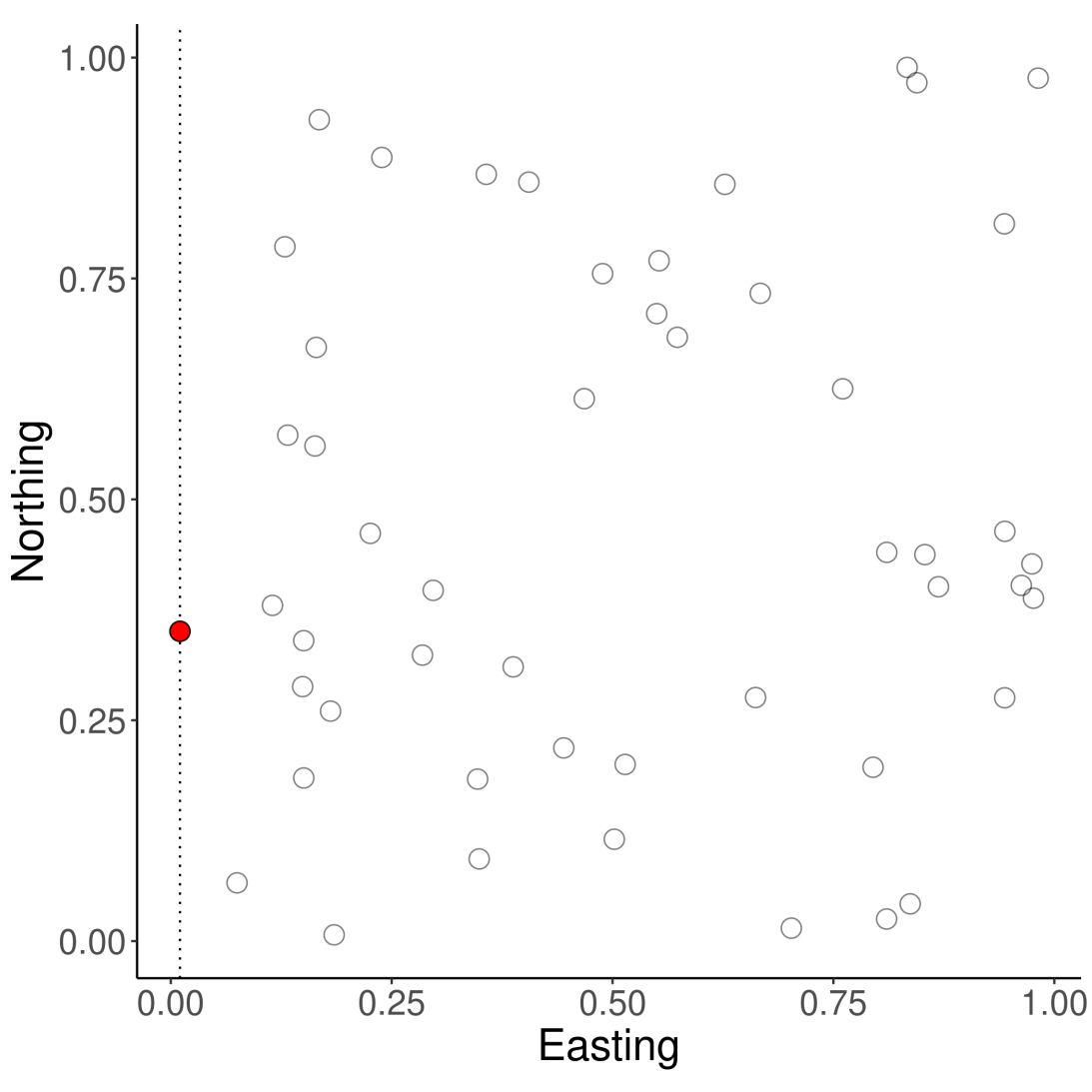
- Spatial covariance functions decay with distance

- 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

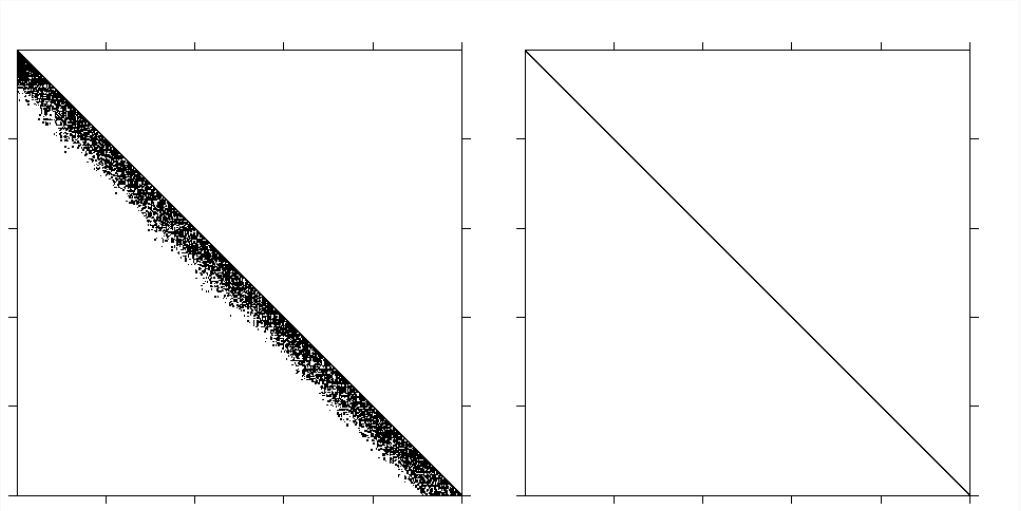
- Other choices proposed in Stein et al. (2004); Gramacy and Apley (2015); Guinness (2018) can also be used, with additional discussion in Finley et al. (2019) and Katzfuzz and Guinness (2021)

Nearest neighbors



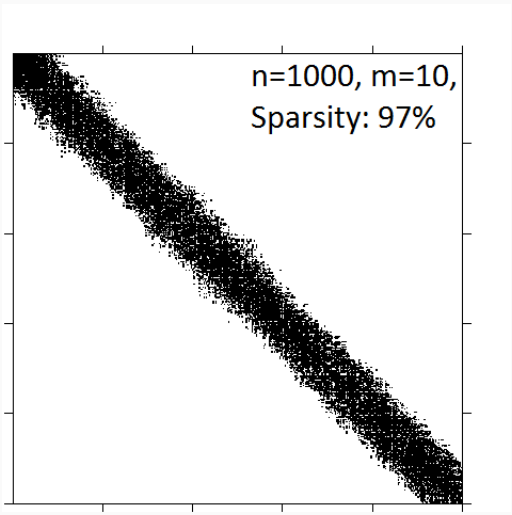
Sparse precision matrices

- 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 $w(\mathbf{s})$

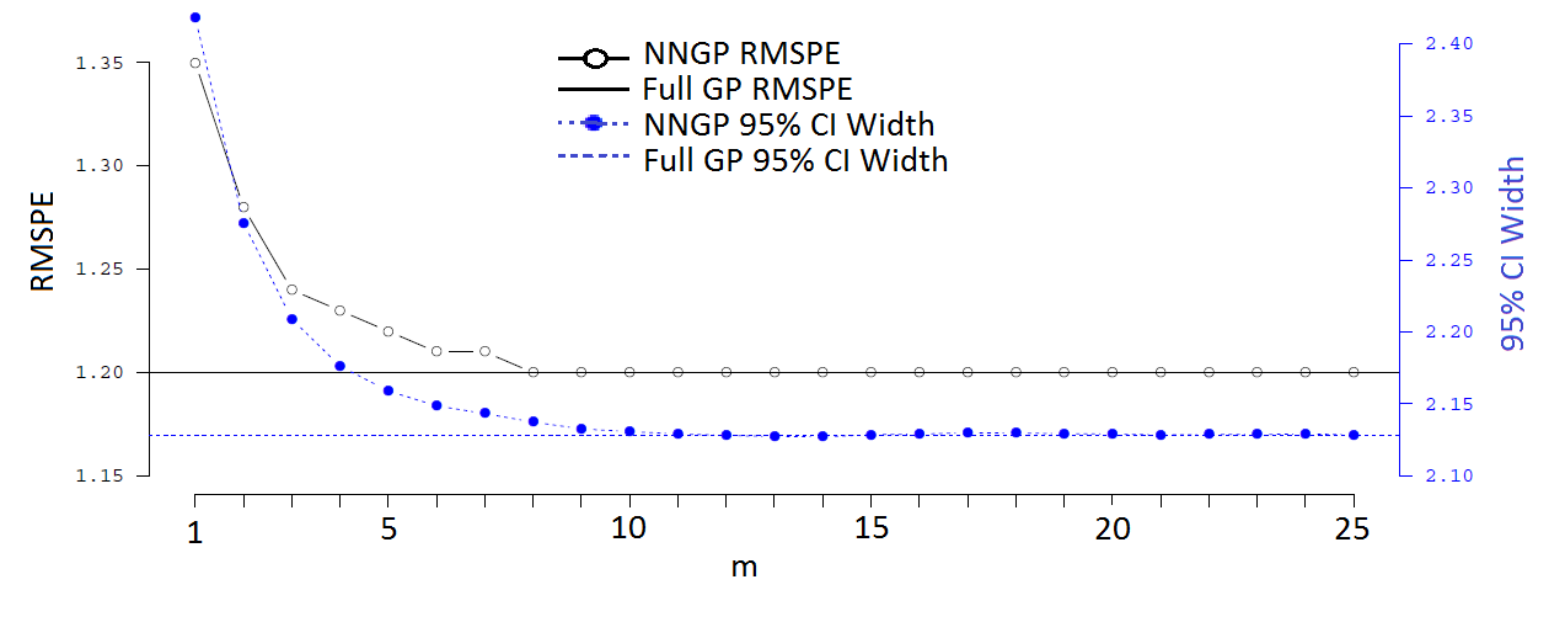
Full Bayesian Model

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

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

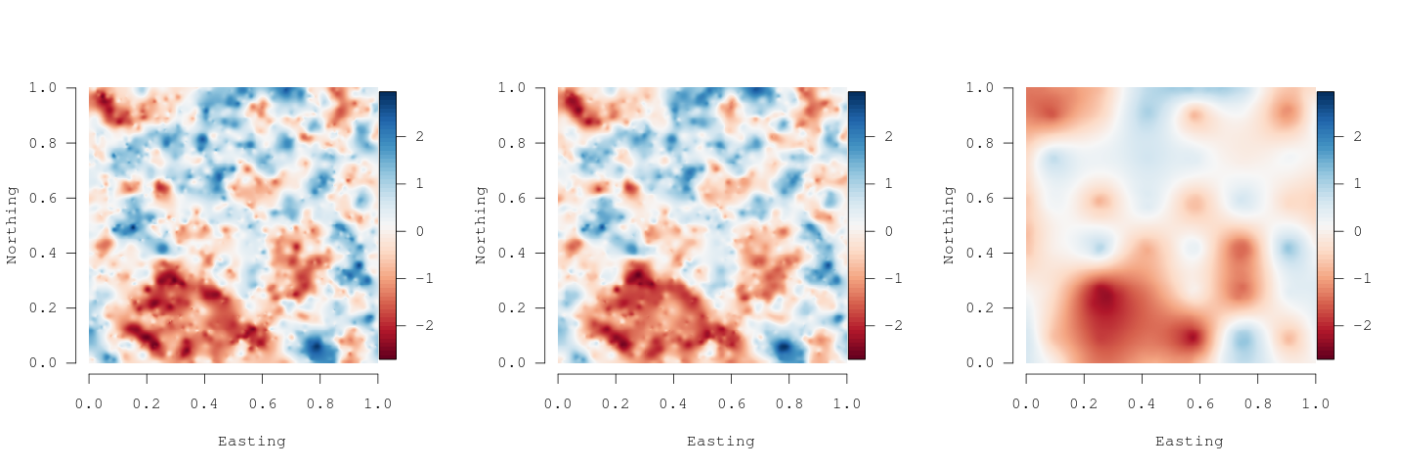
Storage and computation

- 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

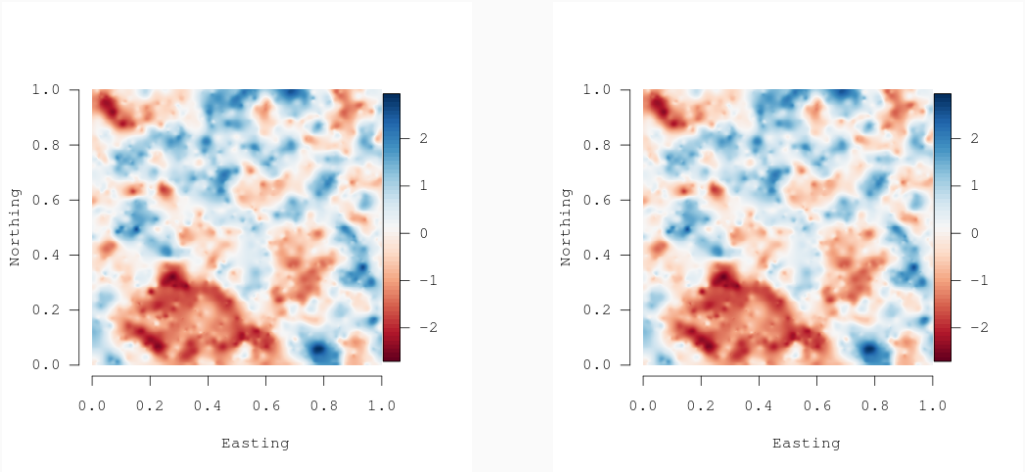
Simulation experiment

- 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

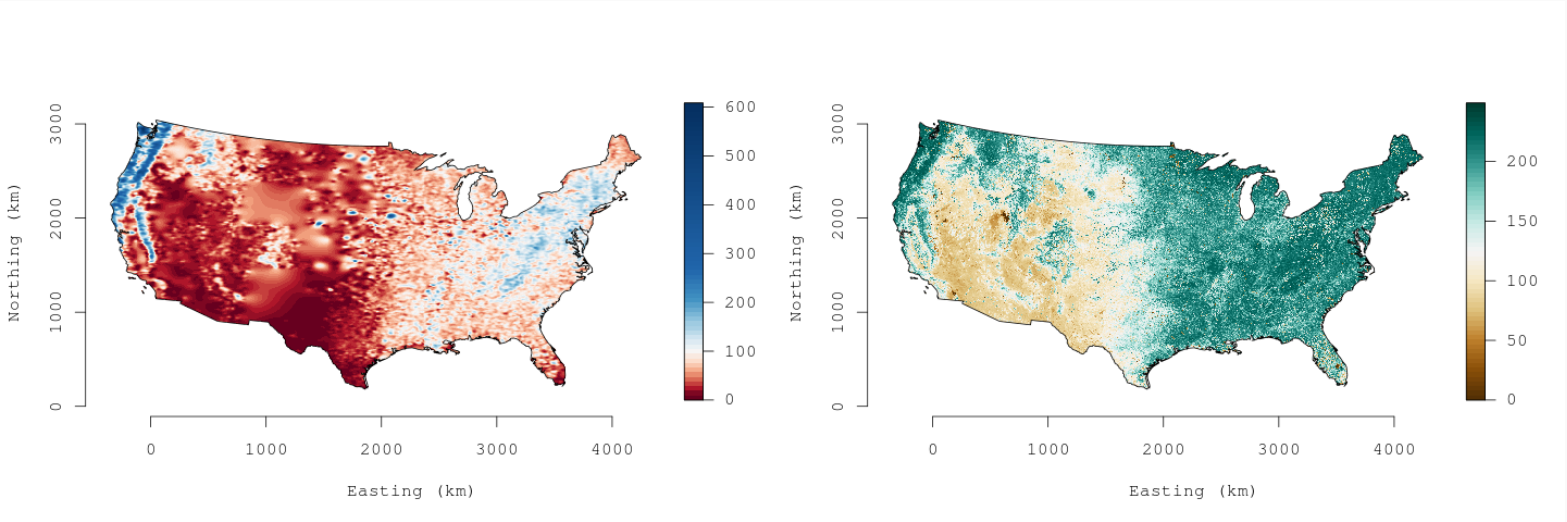
$$\begin{bmatrix} \mathbf{w}(\mathbf{s}_1) \\ \mathbf{w}(\mathbf{s}_2) \\ \mathbf{w}(\mathbf{s}_3) \\ \vdots \\ \mathbf{w}(\mathbf{s}_n) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ \mathbf{A}_{21} & 0 & 0 & \dots & 0 & 0 \\ \mathbf{A}_{31} & \mathbf{A}_{32} & 0 & \dots & 0 & 0 \\ \vdots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

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

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



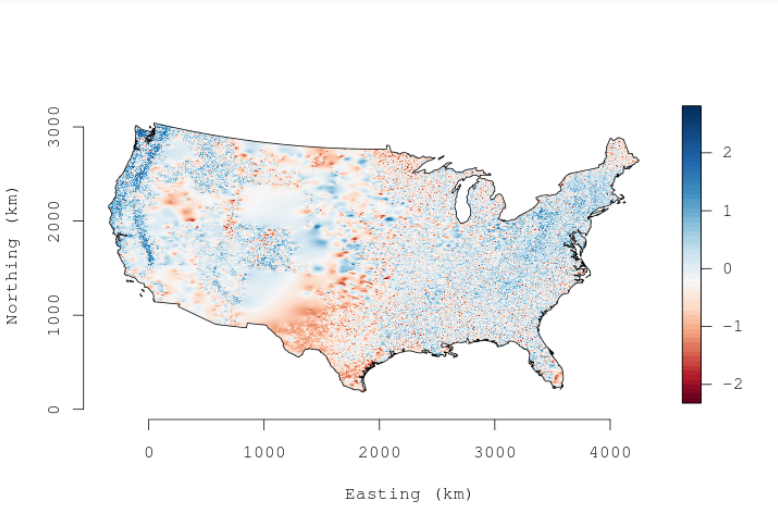
Observed biomass NDVI

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

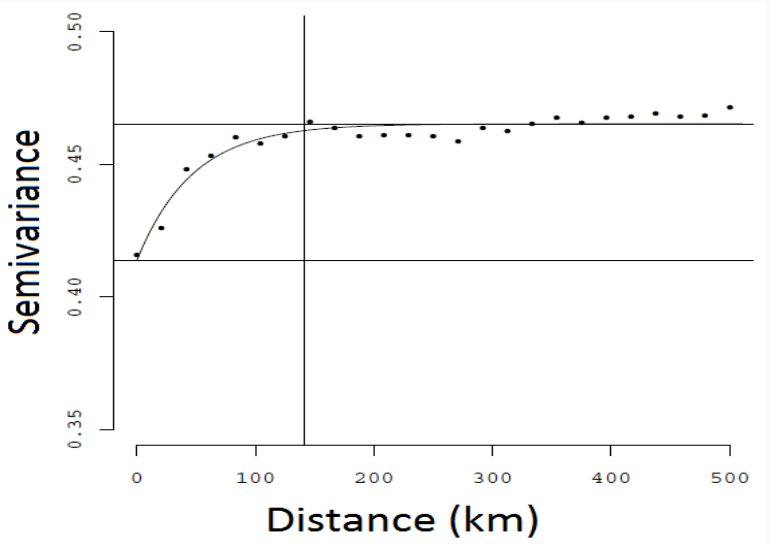
U.S. Forest biomass data

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

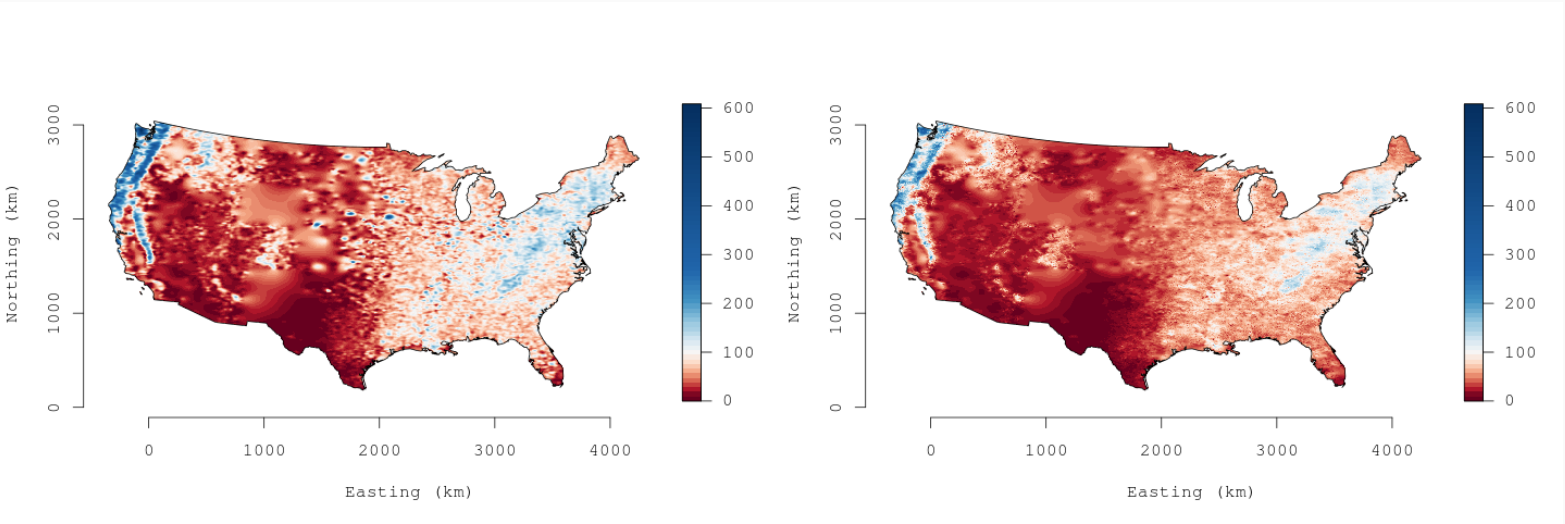
Forest biomass dataset

- $n \approx 10^5$ (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

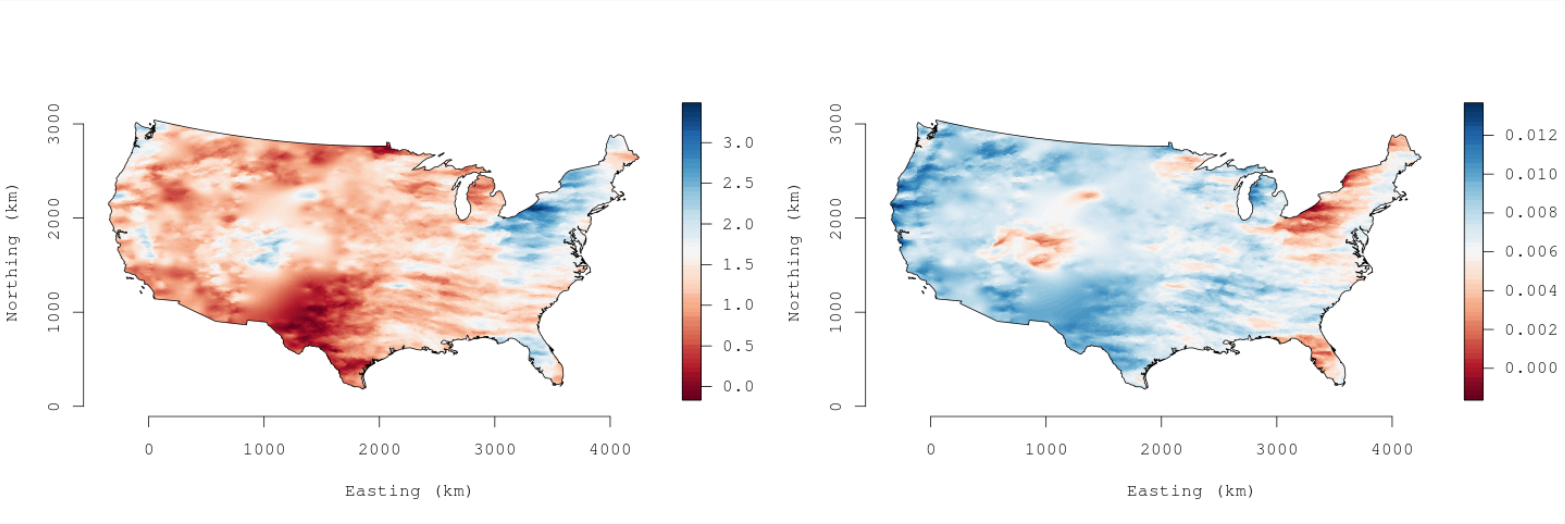
- $Biomass(s) = \beta_0(s) + \beta_1(s) NDVI(s) + \epsilon(s)$
- $\mathbf{w}(s) = (\beta_0(s), \beta_1(s))^\top \sim$ Bivariate $NNGP(0, \tilde{K}(\cdot, \cdot \mid \theta))$, $m = 5$
- Time $\approx$ 6 seconds per iteration
- Full inferential output: 41 hours (25000 MCMC iterations)

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

Comparison of NNGP models

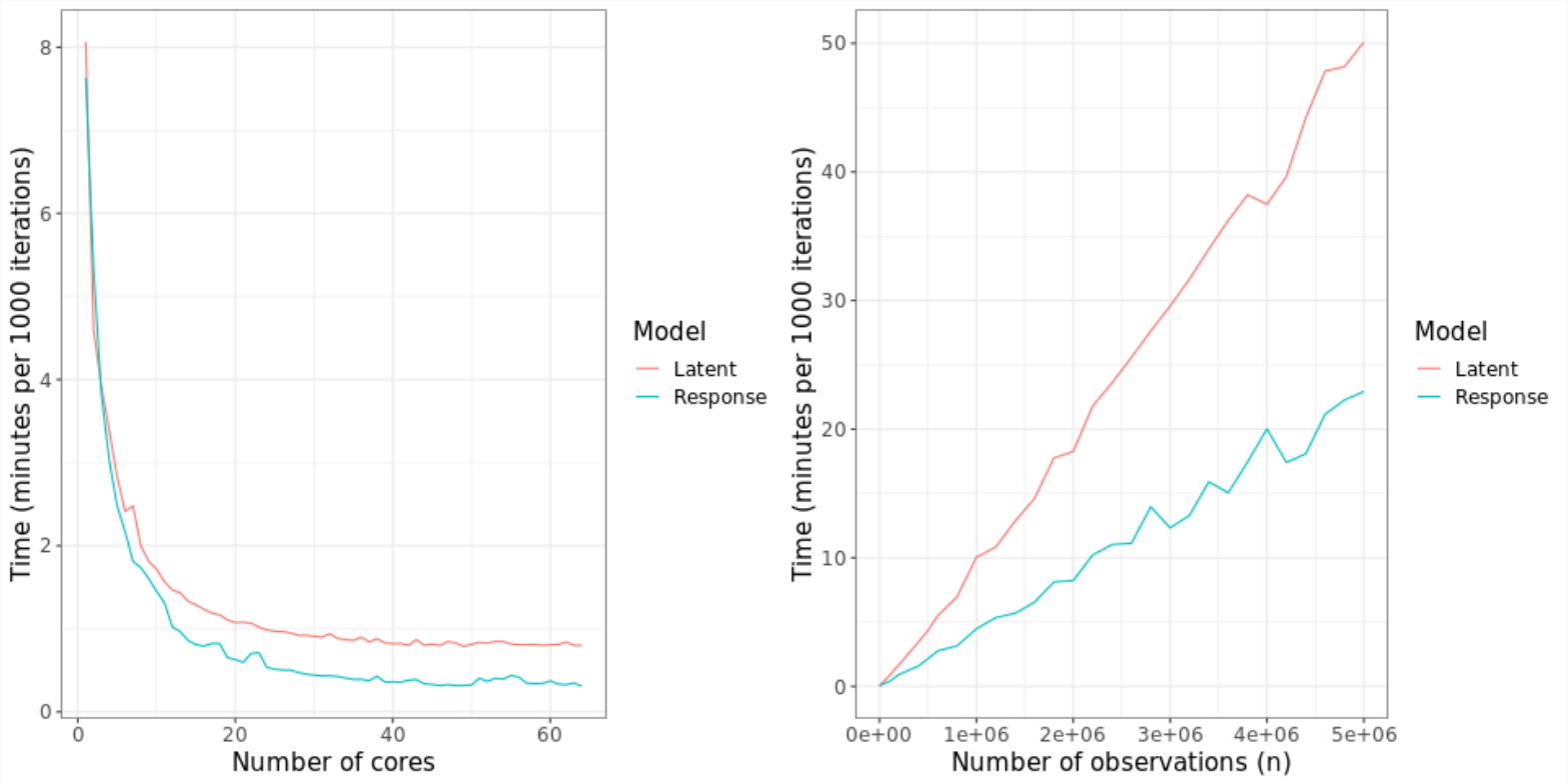


Figure: (a) Runtime for 1000 MCMC iterations for $n = 100000$ and different number of cores. (b) Runtime for 1000 MCMC iterations using 40 cores and n from 1000 to 5 million. Model type (latent and response) refers to different NNGP parameterizations, see Finley et al. 2022.

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

Some notes on efficient computing and high performance computing environments

Andrew Finley[1] & Jeffrey Doser[2]

May 15, 2023

[1]Department of Forestry, Michigan State University.

[2]Department of Integrative Biology, Michigan State University.

Code implementation

Very useful libraries for efficient matrix computation:

1. Fortran BLAS (Basic Linear Algebra Subprograms, see Blackford et al. 2001). Started in late 70s at NASA JPL by Charles L. Lawson. See `http://www.netlib.org/blas`.

2. Fortran LAPACK (Linear Algebra Package, see Anderson et al. 1999). Started in mid 80s at Argonne and Oak Ridge National Laboratories. See `http://www.netlib.org/lapack`.

Modern math software has a heavy reliance on these libraries, e.g., Matlab and *R*. Routines are also accessible via C, C++, Python, etc.

Many improvements on the standard BLAS and LAPACK functions, see, e.g.,

- Intel Math Kernel Library (MKL)
- AMD Core Math Library (ACML)
- Automatically Tuned Linear Algebra Software (ATLAS)
- Matrix Algebra on GPU and Multicore Architecture (MAGMA)
- OpenBLAS `http://www.openblas.net`
- vecLib (for Mac users only)

Key BLAS and LAPACK functions used in our setting.

Function	Description
`dpotrf`	LAPACK routine to compute the Cholesky factorization of a real symmetric positive definite matrix.
`dtrsv`	Level 2 BLAS routine to solve the systems of equations $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{x}$ and $\mathbf{b}$ are vectors and $\mathbf{A}$ is a triangular matrix.
`dtrsm`	Level 3 BLAS routine to solve the matrix equations $\mathbf{AX} = \mathbf{B}$, where $\mathbf{X}$ and $\mathbf{B}$ are matrices and $\mathbf{A}$ is a triangular matrix.
`dgemv`	Level 2 BLAS matrix-vector multiplication.
`dgemm`	Level 3 BLAS matrix-matrix multiplication.

Computing environments

Consider different environments:

1. A distributed system consists of multiple autonomous computers (nodes) that communicate through a network. A computer program that runs in a distributed system is called a distributed program. Message Passing Interface (MPI) is a specification for an Application Programming Interface (API) that allows many computers to communicate.

Consider different environments:

1. A distributed system consists of multiple autonomous computers (nodes) that communicate through a network. A computer program that runs in a distributed system is called a distributed program. Message Passing Interface (MPI) is a specification for an Application Programming Interface (API) that allows many computers to communicate.

2. A shared memory multiprocessing system consists of a single computer with memory that may be simultaneously accessed by one or more programs running on multiple Central Processing Units (CPUs). OpenMP (Open Multi-Processing) is an API that supports shared memory multiprocessing programming.

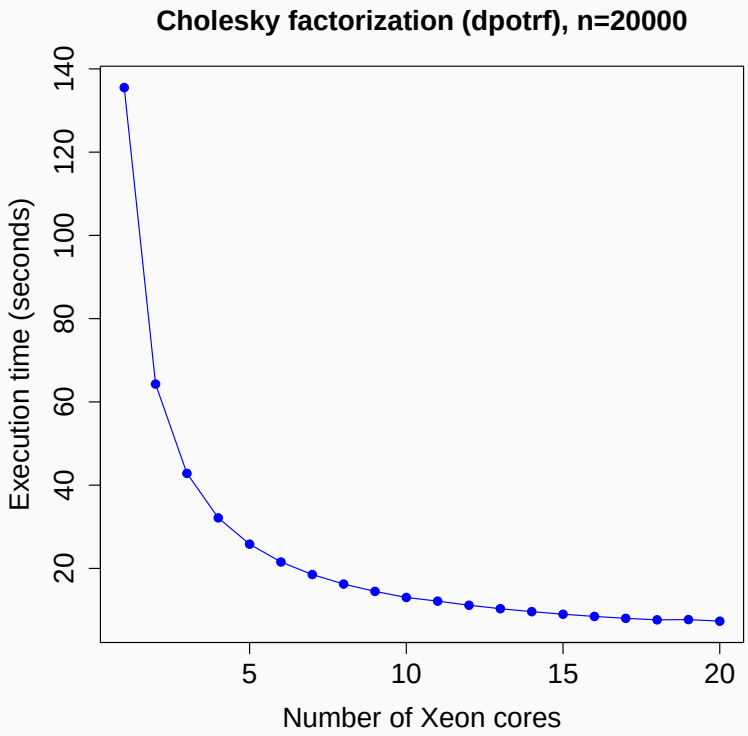
3. A heterogeneous system uses more than one kind of processor, e.g., CPU & (Graphics Processing Unit) GPU or CPU & Intel's Xeon Phi Many Integrated Core (MIC).

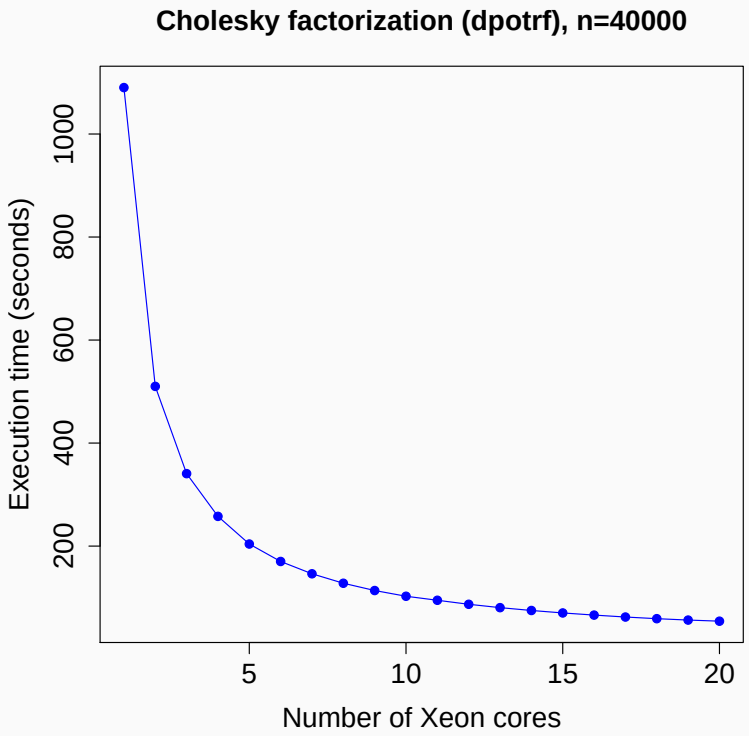
Which environments are right for large n settings?

- MCMC necessitates iterative evaluation of the likelihood which requires operations on large matrices.
- A specific hurdle is factorization to computing determinant and inverse of large dense covariance matrices.
- We try to model our way out and use computing tools to overcome the complexity (e.g., covariance tapering, Kaufman et al. 2008; low-rank methods, Cressie and Johannesson 2008; Banerjee et al. 2008, etc.).
- Due to slow network communication and transport of submatrices among nodes distributed systems are not ideal for these types of iterative large matrix operations.

- My lab currently favors shared memory multiprocessing and heterogeneous systems.
- Newest unit is a Dell Poweredge with 384 GB of RAM, 2 threaded 10-core Xeon CPUs, and 2 Intel Xeon Phi Coprocessor with 61-cores (244 threads) running a Linux operating systems.
- Software includes OpenMP coupled with Intel MKL. MKL is a library of highly optimized, extensively threaded math routines designed for Xeon CPUs and Phi coprocessors (e.g., BLAS, LAPACK, ScaLAPACK, Sparse Solvers, Fast Fourier Transforms, and vector RNGs).

So what kind of speed up to expect from threaded BLAS and LAPACK libraries.





R and threaded BLAS and LAPACK

The BLAS and LAPACK that "ships" with `R` is single-threaded, but these can be replaced with multi-threaded libraries.

Windows

- Microsoft `R` Open: The Enhanced `R` Distribution `https://mran.microsoft.com/open` comes with MLK `https://software.intel.com/en-us/mkl`.
- Replace existing `R`'s `libRblas.so` with OpenBLAS library `libopenblas.so`. OpenBLAS is available here `http://www.openblas.net`.

Max OS X

- Mac vecLib obtained via XCode. Use install notes here.

Linux/Unix

- MKL, OpenBLAS, ACML (compile `R` against MLK or post compile symbolic link of `libRblas.so` to `libopenblas.so`).
- Some additional gains using Intel icc and ifort compilers.

Modeling non-Gaussian spatial data

Jeffrey Doser[1] & Andrew Finley[2]

May 15, 2023

[1]Department of Integrative Biology, Michigan State University.

[2]Department of Forestry, Michigan State University.

Non-Gaussian spatial data

- Often data sets preclude Gaussian modeling: $y(\mathbf{s})$ may not even be continuous
- Examples:
  - Binary: presence or absence of a species at location s.
  - Count: abundance of a species at location s.
  - Categorical: counts of trees by size class at location s.
- Replace Gaussian likelihood by exponential family member (Diggle, Tawn, and Moyeed (1998)).

Hierarchical Bayesian approach

- **First stage**: $y(\mathbf{s}_i)$ are conditionally independent given $\boldsymbol{\beta}$ and $\mathrm{w}(\mathbf{s}_i)$. Here we use a canonical link function, say $g(E[y(\mathbf{s}_i)]) = \eta(\mathbf{s}_i) = \mathbf{x}(\mathbf{s}_i)'\boldsymbol{\beta} + \mathrm{w}(\mathbf{s}_i)$.

Hierarchical Bayesian approach

- **First stage**: $y(\mathbf{s}_i)$ are conditionally independent given $\boldsymbol{\beta}$ and $\mathrm{w}(\mathbf{s}_i)$. Here we use a canonical link function, say $g(E[y(\mathbf{s}_i)]) = \eta(\mathbf{s}_i) = \mathbf{x}(\mathbf{s}_i)'\boldsymbol{\beta} + \mathrm{w}(\mathbf{s}_i)$.
- **Second stage**: Model $\mathrm{w}(\mathbf{s}_i)$ as a Gaussian process:

$$\mathbf{w} \sim N(\mathbf{0}, \sigma^2 \mathbf{R}(\phi))$$

Hierarchical Bayesian approach

- **First stage**: $y(\mathbf{s}_i)$ are conditionally independent given $\boldsymbol{\beta}$ and $\mathrm{w}(\mathbf{s}_i)$. Here we use a canonical link function, say $g(E[y(\mathbf{s}_i)]) = \eta(\mathbf{s}_i) = \mathbf{x}(\mathbf{s}_i)'\boldsymbol{\beta} + \mathrm{w}(\mathbf{s}_i)$.
- **Second stage**: Model $\mathrm{w}(\mathbf{s}_i)$ as a Gaussian process:

$$\mathbf{w} \sim N(\mathbf{0}, \sigma^2 \mathbf{R}(\phi))$$

- **Third stage**: Priors and hyperpriors.

MCMC sampling for spatial GLMMs

- Additional GLMM flexibility comes at a computational cost: lose conjugacy of $\beta, \mathbf{w}$
- Requires more Metroplis steps. Particularly costly for $\mathbf{w}$
- Practical consequence: slower, less efficient algorithms
- Prediction and interpolation proceed as with the Gaussian case

Pólya-Gamma data augmentation

- General approach for Bayesian (spatial) logistic regression that yields conjugate updates of $\boldsymbol{\beta}$ (and $\mathbf{w}$)

Pólya-Gamma data augmentation

- General approach for Bayesian (spatial) logistic regression that yields conjugate updates of β (and $\mathbf{w}$)
- Introduce augmented data $\omega(s_i)$ for each $i = 1, \ldots, n$, where $\omega(s_i) \sim \text{PG}(N(\mathbf{s}_i), 0)$, with $N(\mathbf{s}_i)$ the Binomial weights

Pólya-Gamma data augmentation

- General approach for Bayesian (spatial) logistic regression that yields conjugate updates of β (and $\mathbf{w}$)
- Introduce augmented data $\omega(s_i)$ for each $i = 1, \ldots, n$, where $\omega(s_i) \sim \text{PG}(N(\mathbf{s}_i), 0)$, with $N(\mathbf{s}_i)$ the Binomial weights
- Define $\kappa(\mathbf{s}_i) = y(\mathbf{s}_i) - N(\mathbf{s}_i)/2$

Pólya-Gamma data augmentation

- General approach for Bayesian (spatial) logistic regression that yields conjugate updates of β (and $\mathbf{w}$)
- Introduce augmented data $\omega(s_i)$ for each $i = 1, \ldots, n$, where $\omega(s_i) \sim \text{PG}(N(\mathbf{s}_i), 0)$, with $N(\mathbf{s}_i)$ the Binomial weights
- Define $\kappa(\mathbf{s}_i) = y(\mathbf{s}_i) - N(\mathbf{s}_i)/2$
- Resulting Gibbs sampler is remarkably similar to that of a Gaussian model with response $y(\mathbf{s}_i)^* = \kappa(\mathbf{s}_i)/\omega(\mathbf{s}_i)$ and heteroskedastic variances $\tau^2(\mathbf{s}_i) = 1/\omega(\mathbf{s}_i)$.

Pólya-Gamma data augmentation

- Suppose $y(\mathbf{s}_i) \sim \text{Bernoulli}(\psi(\mathbf{s}_i))$.

$$\psi(\mathbf{s}_i)^{y(\mathbf{s}_i)}(1-\psi(\mathbf{s}_i))^{1-y(\mathbf{s})} = \frac{\exp(\mathbf{x}(\mathbf{s}_i)^\top \boldsymbol{\beta} + \mathrm{w}(\mathbf{s}_i))^{y(\mathbf{s}_i)}}{1+\exp(\mathbf{x}_i^\top \boldsymbol{\beta} + \mathrm{w}(\mathbf{s}_i))}$$
$$= \exp(\kappa(\mathbf{s}_i)(\mathbf{x}(\mathbf{s}_i)^\top \boldsymbol{\beta} + \mathrm{w}(\mathbf{s}_i)))\times$$
$$\int \exp(-\frac{\omega(\mathbf{s}_i)}{2}(\mathbf{x}(\mathbf{s}_i)^\top \boldsymbol{\beta} + \mathrm{w}(\mathbf{s}_i)))^2 \times$$
$$p(\omega(\mathbf{s}_i) \mid 1, 0) d\omega(\mathbf{s}_i),$$

- $p(\omega(\mathbf{s}_i) \mid 1, 0)$ is the Pólya-Gamma PDF with parameters 1 and 0
- With Gaussian priors on $\boldsymbol{\beta}$ and IG prior on σ^2, full conditionals for $\boldsymbol{\beta}$, σ^2, and $\mathbf{w}$ are available in closed form. ϕ updated with MH.
- **See Polson, Scott, Windle (2013) JASA**

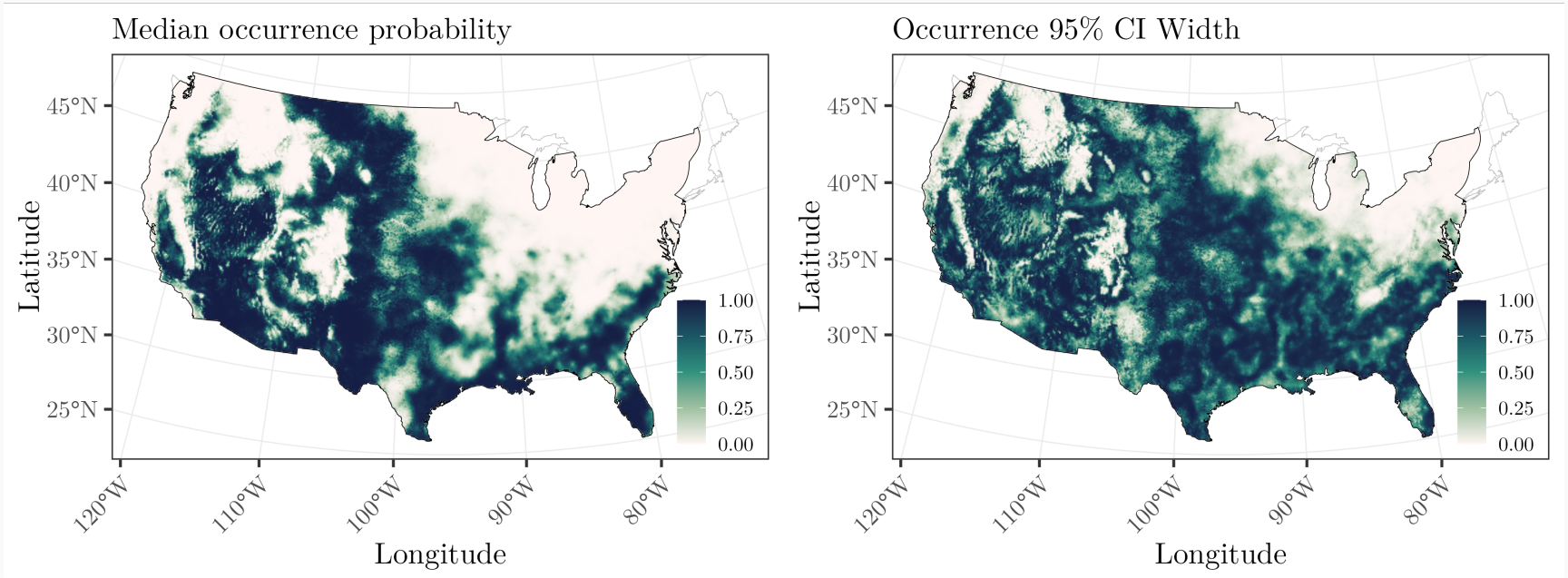
Example: Binary spatial regression

- Objective: predict the distribution of Loggerhead Shrike across the US

$$
\begin{aligned}
y(\mathbf{s}_i) &\sim \text{Bernoulli}(\psi(\mathbf{s}_i)) \\
\text{logit}(\psi(\mathbf{s}_i)) &= \mathbf{x}(\mathbf{s}_i)^\top \boldsymbol{\beta} + w(\mathbf{s}_i) \\
\mathbf{w} &\sim N(\mathbf{0}, \sigma^2 \mathbf{R}(\phi)) \\
\boldsymbol{\beta} &\sim N(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta) \\
\sigma^2 &\sim IG(a_\sigma, b_\sigma) \\
\phi &\sim \text{Uniform}(l, u) \\
\omega(\mathbf{s}_i) &\sim \text{PG}(1, 0)
\end{aligned}
$$

Example: Binary spatial regression

Posterior predictive inference proceeds as with the Gaussian case



Some practical considerations

- Priors for σ^2 and ϕ may need to be more informative, particularly for binary data.

Some practical considerations

- Priors for σ^2 and ϕ may need to be more informative, particularly for binary data.
- Be careful with non-identity link functions when thinking about priors.

Some practical considerations

- Priors for σ^2 and ϕ may need to be more informative, particularly for binary data.
- Be careful with non-identity link functions when thinking about priors.
- Pólya-Gamma data augmentation works really for binomial data. Computational cost increases as Binomial weights increases.

Some practical considerations

- Priors for σ^2 and ϕ may need to be more informative, particularly for binary data.
- Be careful with non-identity link functions when thinking about priors.
- Pólya-Gamma data augmentation works really for binomial data. Computational cost increases as Binomial weights increases.
- Pólya-Gamma data augmentation also applicable for Negative Binomial count data, but slow for large counts and can be unstable.

Software

- `spBayes`
  - Univariate and multivariate, full GPs or predictive processes
  - Gaussian, Binomial (no Pólya-Gamma data augmentation), Poisson
- `spNNGP`
  - Univariate, NNGPs
  - Gaussian, Binomial
- `spOccupancy`
  - Univariate and multivariate, focus on modeling wildlife distributions, full GPs or NNGPs
  - Bernoulli
- `spAbundance` (`https://github.com/doserjef/spAbundance`)
  - Univariate and multivariate, focus on modeling wildlife/plant abundance, NNGPs
  - Gaussian, Poisson, Negative Binomial

Spatial Factor Models for Multivariate Spatial Data

Jeffrey Doser[1] & Andrew Finley[2]

May 15, 2023

[1]Department of Integrative Biology, Michigan State University.

[2]Department of Forestry, Michigan State University.

Multivariate spatial data

- Point-referenced spatial data often come as multivariate measurements at each location.

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 $PM_{2.5}$.

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 $PM_{2.5}$.
  - Community Ecology: assemblages/communities of species

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 $PM_{2.5}$.
  - Community Ecology: assemblages/communities of species
  - Forestry: measurements of stand characteristics age, total biomass, and average tree diameter.

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 $PM_{2.5}$.
  - Community Ecology: assemblages/communities of species
  - 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

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 $PM_{2.5}$.
  - Community Ecology: assemblages/communities of species
  - 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

Multivariate spatial generalized linear model

- Spatial generalized linear model for h-variate spatial data for $j = 1, 2, \ldots, h$ and $i = 1, \ldots, n$:

$$y_j(\mathbf{s}_i) \sim f(\mu_j(\mathbf{s}_i), \tau_j)$$
$$\mu_j(\mathbf{s}_i) = g^{-1}(\eta_j(\mathbf{s}_i)) = \mathbf{x}(\mathbf{s}_i)^\top \boldsymbol{\beta}_j + \mathrm{w}_j^*(\mathbf{s}_i)$$

- We can imagine modeling $\mathbf{w}^*(\mathbf{s}_i) = (\mathrm{w}_1^*(\mathbf{s}_i), \mathrm{w}_2^*(\mathbf{s}_i), \ldots, \mathrm{w}_h^*(\mathbf{s}_i))'$ as an h-variate Gaussian process

Multivariate spatial generalized linear model

- Spatial generalized linear model for h-variate spatial data for $j = 1, 2, \ldots, h$ and $i = 1, \ldots, n$:

$$y_j(\mathbf{s}_i) \sim f(\mu_j(\mathbf{s}_i), \tau_j)$$
$$\mu_j(\mathbf{s}_i) = g^{-1}(\eta_j(\mathbf{s}_i)) = \mathbf{x}(\mathbf{s}_i)^\top \boldsymbol{\beta}_j + \mathrm{w}_j^*(\mathbf{s}_i)$$

- We can imagine modeling $\mathbf{w}^*(\mathbf{s}_i) = (\mathrm{w}_1^*(\mathbf{s}_i), \mathrm{w}_2^*(\mathbf{s}_i), \ldots, \mathrm{w}_h^*(\mathbf{s}_i))'$ as an h-variate Gaussian process
- Could model using Multivariate NNGP as disussed previously with SVCs, works well when $h < 5$.

Multivariate spatial generalized linear model

- Spatial generalized linear model for h-variate spatial data for $j = 1, 2, \ldots, h$ and $i = 1, \ldots, n$:

$$y_j(\mathbf{s}_i) \sim f(\mu_j(\mathbf{s}_i), \tau_j)$$
$$\mu_j(\mathbf{s}_i) = g^{-1}(\eta_j(\mathbf{s}_i)) = \mathbf{x}(\mathbf{s}_i)^\top \boldsymbol{\beta}_j + \mathrm{w}_j^*(\mathbf{s}_i)$$

- We can imagine modeling $\mathbf{w}^*(\mathbf{s}_i) = (\mathrm{w}_1^*(\mathbf{s}_i), \mathrm{w}_2^*(\mathbf{s}_i), \ldots, \mathrm{w}_h^*(\mathbf{s}_i))'$ as an h-variate Gaussian process
- Could model using Multivariate NNGP as disussed previously with SVCs, works well when $h < 5$.
- But what about when h is large (e.g,. 10, 100)?

Spatial Factor Model

- Approximates the dependence between multivariate (spatially-dependent) outcomes through a linear combination of a (much) lower-dimensional set of spatial factors

Spatial Factor Model

- Approximates the dependence between multivariate (spatially-dependent) outcomes through a linear combination of a (much) lower-dimensional set of spatial factors
- We represent the $h \times 1$ vector $\mathbf{w}^*(\mathbf{s}_i)$ as a linear combination of latent spatial factors and factor loadings:

$$\mathbf{w}^*(\mathbf{s}_i) = \mathbf{\Lambda}\mathbf{w}(\mathbf{s}_i)$$

- $\mathbf{\Lambda}$ is an $h \times q$ loadings matrix (tall and skinny) and $\mathbf{w}(\mathbf{s}_i)$ is a $q \times 1$ vector of realizations from q *independent* spatial GPs

Spatial Factor Model

- Approximates the dependence between multivariate (spatially-dependent) outcomes through a linear combination of a (much) lower-dimensional set of spatial factors
- We represent the $h \times 1$ vector $\mathbf{w}^*(\mathbf{s}_i)$ as a linear combination of latent spatial factors and factor loadings:

$$\mathbf{w}^*(\mathbf{s}_i) = \mathbf{\Lambda}\mathbf{w}(\mathbf{s}_i)$$

- $\mathbf{\Lambda}$ is an $h \times q$ loadings matrix (tall and skinny) and $\mathbf{w}(\mathbf{s}_i)$ is a $q \times 1$ vector of realizations from q *independent* spatial GPs
- In traditional factor analysis, $\mathbf{w}(\mathbf{s}_i)$ are realizations from independent standard normal random variables.

Spatial Factor Model

- Choosing $q << m$ leads to substantial computational reductions.
- Simple to code: just sample from q independent GPs as with basic univariate models.
- Yields a non-separable multivariate cross-covariance function given by $\sum_{k=1}^{q} \mathbf{R}_k(\phi_k)\boldsymbol{\lambda}_k\boldsymbol{\lambda}_k^\top$
- Can simply replace the q full GPs with their corresponding NNGPs to yield a spatial factor NNGP model
- Identifiability constraints on $\boldsymbol{\Lambda}$: fix upper triangle to 0 and diagonal to 1. See Ren and Banerjee (2013) *Biometrics*

Priors

- Standard normal priors for the lower triangle of $\boldsymbol{\Lambda}$
- We like to model response-specific regression coefficients $\boldsymbol{\beta}_j$ hierarchically. For each $r = 1, \ldots, p$ covariate, we model $\beta_{j,r}$ following

$$\beta_{j,r} \sim N(\mu_{\beta_r}, \tau^2_{\beta_r})$$

- Gaussian hyperpriors for μ_{β_r} and IG or half-Cauchy priors for $\tau^2_{\beta_r}$
- Independent uniform priors for spatial decay parameters ϕ

Gibbs sampler

- Full conditionals are in closed form for all parameters except ϕ.
- Update ϕ with an Adaptive Metroplis-within-Gibbs algorithm (Roberts and Rosenthal 2009)
- See Taylor-Rodriguez et al. 2019 for Gaussian sampler, spOccupancy website for Pólya-Gamma sampler

Why we like spatial factor models

- Simple to code (don't need to deal with cross-covariance matrices).

Why we like spatial factor models

- Simple to code (don't need to deal with cross-covariance matrices).
- Relatively fast and efficient (well, at least for Gaussian and Binomial).

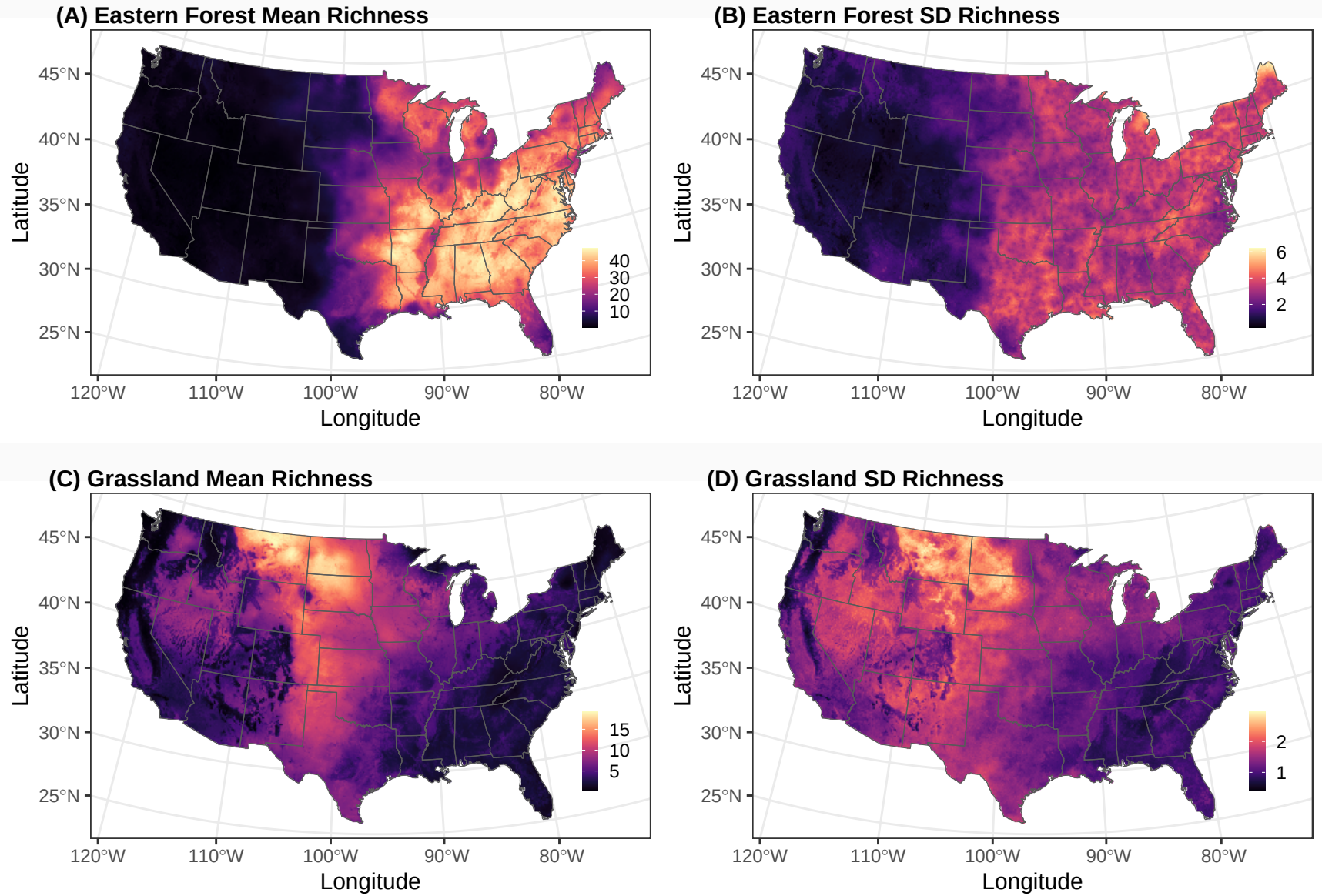
Why we like spatial factor models

- Simple to code (don't need to deal with cross-covariance matrices).
- Relatively fast and efficient (well, at least for Gaussian and Binomial).
- Factors and factor loadings can be used for model-based ordination.

Why we like spatial factor models

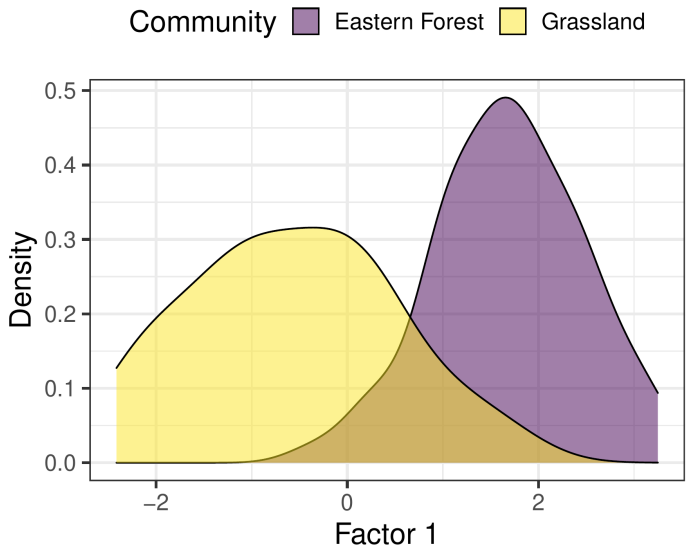
- Simple to code (don't need to deal with cross-covariance matrices).
- Relatively fast and efficient (well, at least for Gaussian and Binomial).
- Factors and factor loadings can be used for model-based ordination.
- Straightforward extensions to spatially-varying coefficient models.

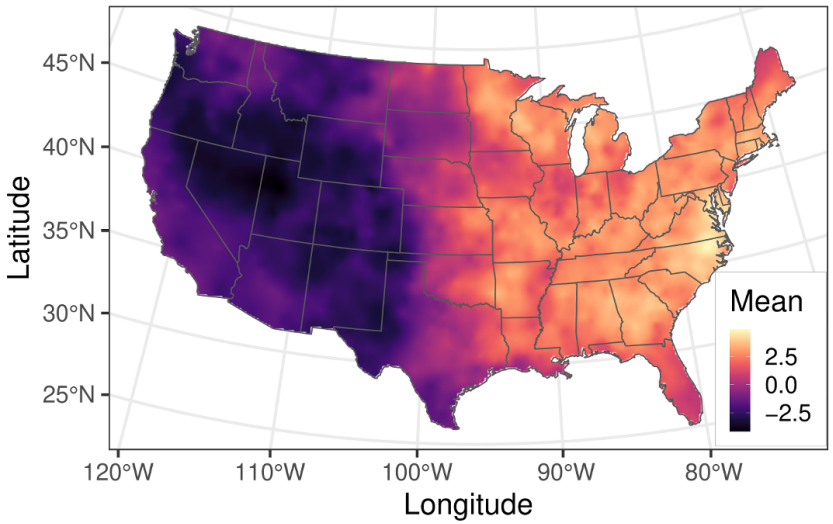
Example: bird communities across the continental US



Example: bird communities across the continental US

Visualization of the first spatial factor and corresponding factor loadings





Some downsides to spatial factor models

- Convergence assessment is not always straightforward
- Sensitivity to initial values
- Order of the first q species has important implications for convergence and mixing.
- Assume a multivariate stochastic process can be represented as a linear combination of independent univariate processes

Software

- `spOccupancy`: spatial NNGP and non-spatial factor models for binary data
- `spAbundance`: Gaussian, Poisson, and NB spatial NNGP and non-spatial factor models.
- `boral`: many distributions for non-spatial and spatial factor models (Hui 2015 *MEE*; spatial use full GPs fit in JAGS)
- `Hmsc`: spatial models using NNGPs (Tikhonov et al. 2019; *MEE*)
- `spBFA`: a variety of spatial models with some nifty priors (Berchuck et al. 2022 *Bayesian Analysis*)

Exercise

Modeling the distribution of 10 tree species across Vermont

Application of Spatially-Varying Coefficient Models

Jeffrey Doser[1] & Andrew Finley[2]

May 15, 2023

[1]Department of Integrative Biology, Michigan State University.

[2]Department of Forestry, Michigan State University.

Spatially-Varying Coefficient Models

- Extension of spatial regression approaches that allow regression coefficients to vary across space, and not just the intercept
- SVC models are random slopes models, with spatial structure given to the random slopes

SVC GLMMs

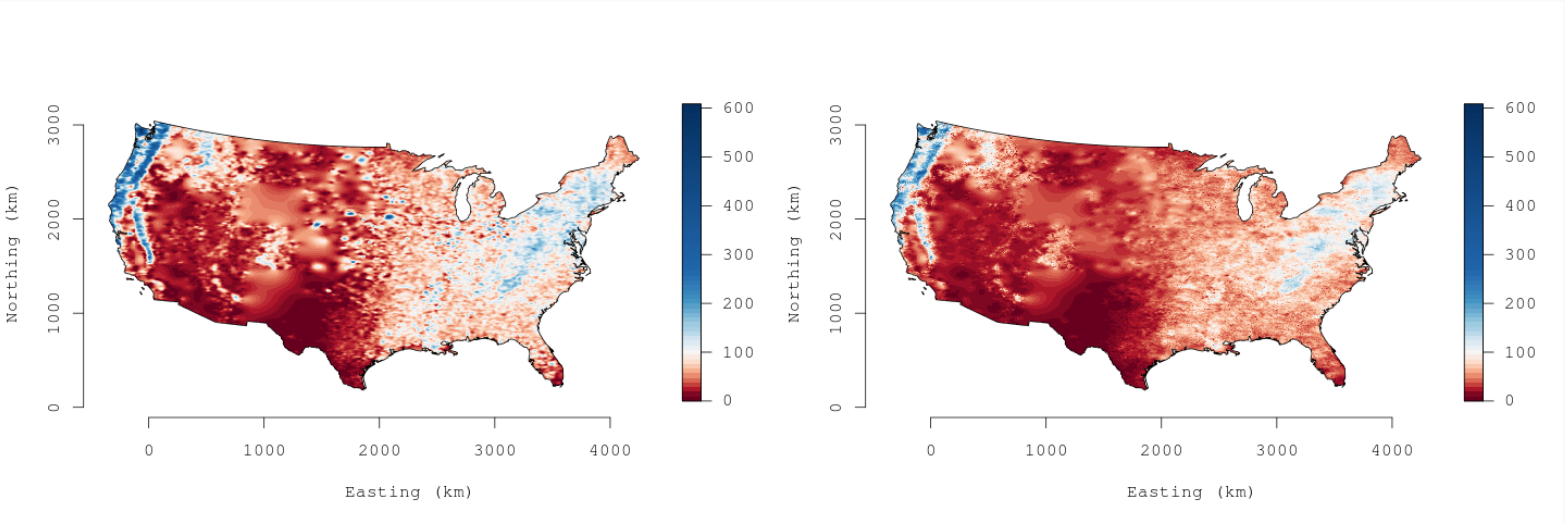
$$
\begin{aligned}
y(\mathbf{s}_i) &\sim f(\mu(\mathbf{s}_i), \tau) \\
\mu(\mathbf{s}_i) &= g^{-1}(\eta(\mathbf{s}_i)) = \mathbf{x}(\mathbf{s}_i)^\top \tilde{\boldsymbol{\beta}}(\mathbf{s}_i) \\
\tilde{\beta}_r(\mathbf{s}_i) &= \beta_r + \mathrm{w}_r(\mathbf{s}_i) \text{ for each } r = 1, \ldots, p
\end{aligned}
$$

- We can model $\mathbf{w}(\mathbf{s}_i)$ using a GP, predictive process, or NNGP
- We can envision modeling $\mathbf{w}(\mathbf{s}_i)$ in two ways:
  1. Multivariate NNGP (see previous forest biomass example)
  2. Independent NNGPs
- Here we focus on the latter
- Pros and cons to both approaches, similar to correlations between random slopes and intercepts in mixed models

Potential benefits of SVC models

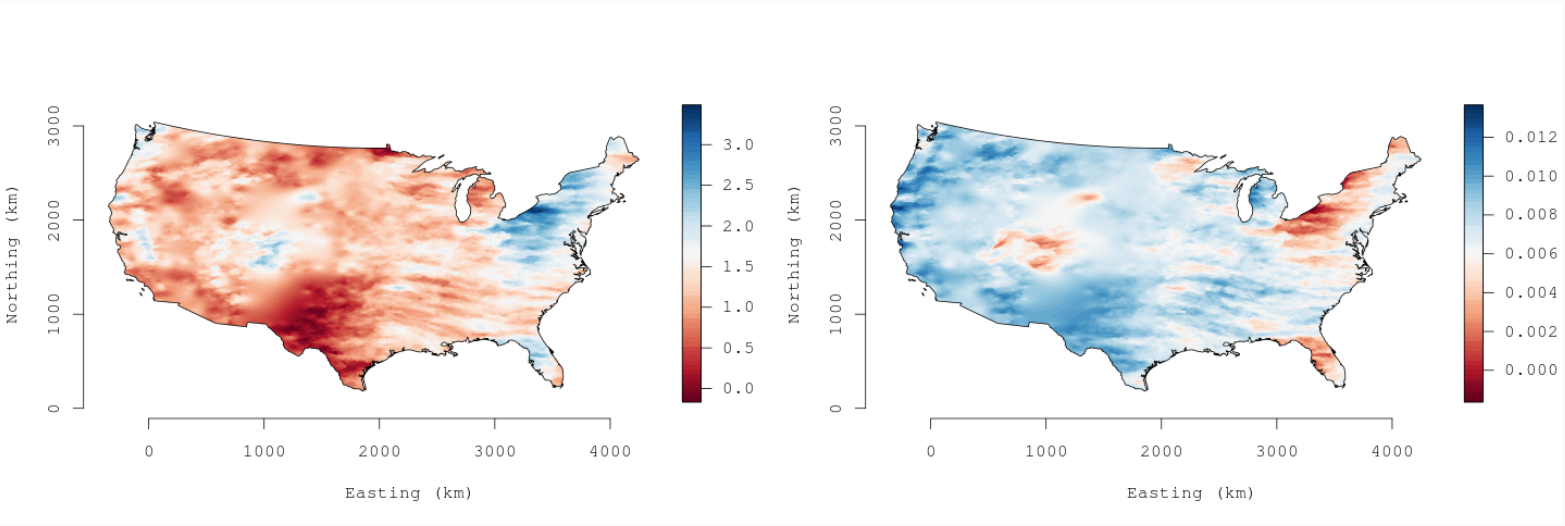
- Improved predictive performance
- Tremendous flexibility to accommodate spatial variability in effects
- Hypothesis testing and generation
- Accommodate highly non-linear relationships
- Model spatial variability in trends over time

Improved predictive performance



Observed biomass

Fitted biomass

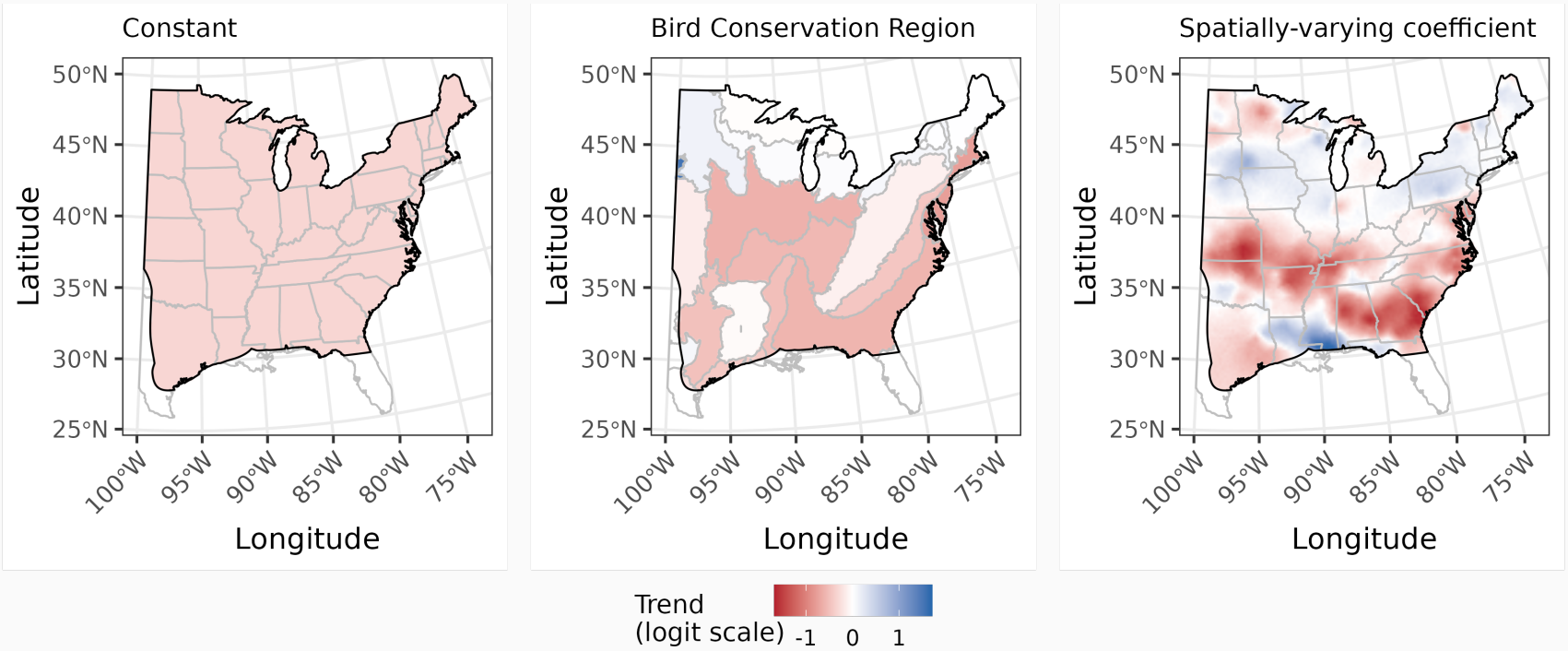


$\beta_0(s)$

$\beta_{NDVI}(s)$

More flexibility to accommodate spatial variability in effects

Gray Catbird occurrence trend across the eastern US from 2000-2019



Some cautionary notes: (1) Standardization of covariates

- Common recommendation to center and scale covariates is likely not appropriate. Why?

Some cautionary notes: (1) Standardization of covariates

- Common recommendation to center and scale covariates is likely not appropriate. Why?
- Consider a single SVC $w(\mathbf{s}_i)$:

Some cautionary notes: (1) Standardization of covariates

- Common recommendation to center and scale covariates is likely not appropriate. Why?
- Consider a single SVC $w(\mathbf{s}_i)$:
  - $\text{var}(w(\mathbf{s}_i)) = x^2(\mathbf{s}_i)\sigma^2$

Some cautionary notes: (1) Standardization of covariates

- Common recommendation to center and scale covariates is likely not appropriate. Why?
- Consider a single SVC $w(\mathbf{s}_i)$:
  - $\text{var}(w(\mathbf{s}_i)) = x^2(\mathbf{s}_i)\sigma^2$
  - $\text{cov}(w(\mathbf{s}_i), w(\mathbf{s}_j)) = \sigma^2 x(\mathbf{s}_i)x(\mathbf{s}_j)\rho(|\ \mathbf{s}_i - \mathbf{s}_j\ |; \phi)$

Some cautionary notes: (1) Standardization of covariates

- Common recommendation to center and scale covariates is likely not appropriate. Why?
- Consider a single SVC $w(\mathbf{s}_i)$:
  - $\text{var}(w(\mathbf{s}_i)) = x^2(\mathbf{s}_i)\sigma^2$
  - $\text{cov}(w(\mathbf{s}_i), w(\mathbf{s}_j)) = \sigma^2 x(\mathbf{s}_i)x(\mathbf{s}_j)\rho(\mid \mathbf{s}_i - \mathbf{s}_j \mid; \phi)$
  - If x takes both positive and negative values, variance will be high at large and small x values and small around zero.

Some cautionary notes: (1) Standardization of covariates

- Common recommendation to center and scale covariates is likely not appropriate. Why?
- Consider a single SVC $w(\mathbf{s}_i)$:
  - $\text{var}(w(\mathbf{s}_i)) = x^2(\mathbf{s}_i)\sigma^2$
  - $\text{cov}(w(\mathbf{s}_i), w(\mathbf{s}_j)) = \sigma^2 x(\mathbf{s}_i)x(\mathbf{s}_j)\rho(\mid \mathbf{s}_i - \mathbf{s}_j \mid; \phi)$
  - If x takes both positive and negative values, variance will be high at large and small x values and small around zero.
  - Further, negative covariances can arise, which may drive σ^2 to be very close to zero.

Some cautionary notes: (1) Standardization of covariates

- Common recommendation to center and scale covariates is likely not appropriate. Why?
- Consider a single SVC $\mathrm{w}(\mathbf{s}_i)$:
  - $\mathrm{var}(\mathrm{w}(\mathbf{s}_i)) = \mathrm{x}^2(\mathbf{s}_i)\sigma^2$
  - $\mathrm{cov}(\mathrm{w}(\mathbf{s}_i), \mathrm{w}(\mathbf{s}_j)) = \sigma^2 \mathrm{x}(\mathbf{s}_i)\mathrm{x}(\mathbf{s}_j)\rho(\mid \mathbf{s}_i - \mathbf{s}_j \mid; \phi)$
  - If x takes both positive and negative values, variance will be high at large and small x values and small around zero.
  - Further, negative covariances can arise, which may drive σ^2 to be very close to zero.
- Recommendation following Gelfand et al. (2003) *JASA*: only use positive covariate values. Can lead to identifiability problems.
- For modeling spatially-varying trends, standardiziation is recommended.

Some cautionary notes: (2) Inference

- Estimating multiple GPs simultaneously is difficult.
- Use caution when making inference from SVC models, particularly with observational data.

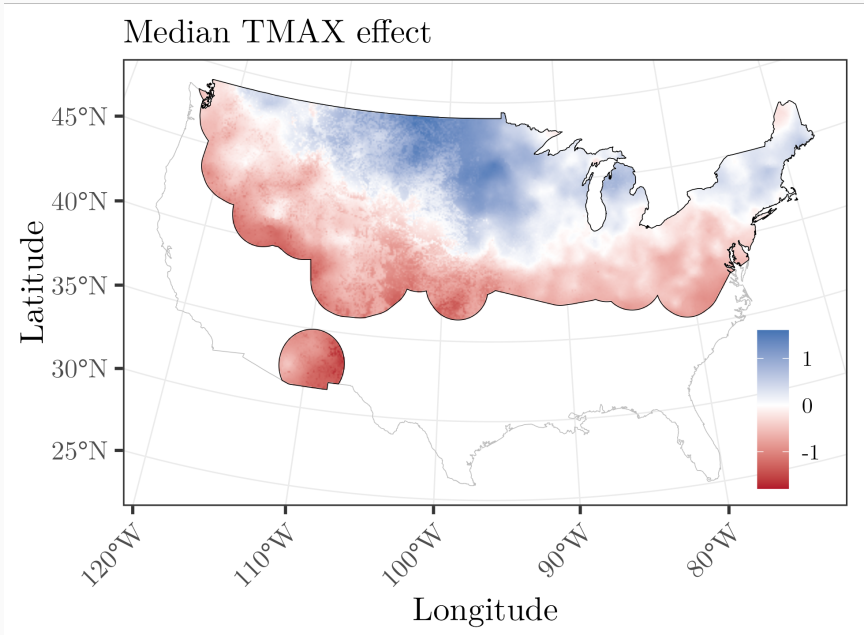
Some cautionary notes: (2) Inference

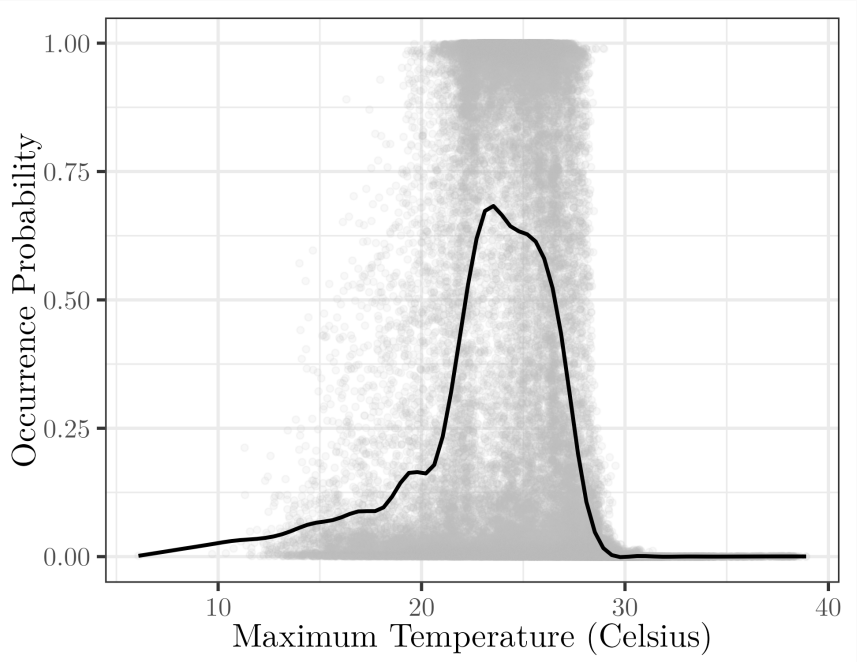
- Estimating multiple GPs simultaneously is difficult.
- Use caution when making inference from SVC models, particularly with observational data.
- The model only identifies the product of $\tilde{\beta}(\mathbf{s}_i)$ and $x(\mathbf{s}_i)$. Prediction at $\mathbf{s}_i$ with a new covariate value (e.g., $x^*(\mathbf{s}_i)$) may not be appropriate.

Some cautionary notes: (2) Inference

- Estimating multiple GPs simultaneously is difficult.
- Use caution when making inference from SVC models, particularly with observational data.
- The model only identifies the product of $\tilde{\beta}(\mathbf{s}_i)$ and $x(\mathbf{s}_i)$. Prediction at $\mathbf{s}_i$ with a new covariate value (e.g., $x^*(\mathbf{s}_i)$) may not be appropriate.
- Recommend plotting spatial maps of SVC effects together with a plot of the covariate vs. the predicted mean for all observed locations
- Interpretation can be difficult when both the effect and the covariate vary over space. More straightforward with something like a trend.

Example: Effect of max temperature on Bobolink occurrence



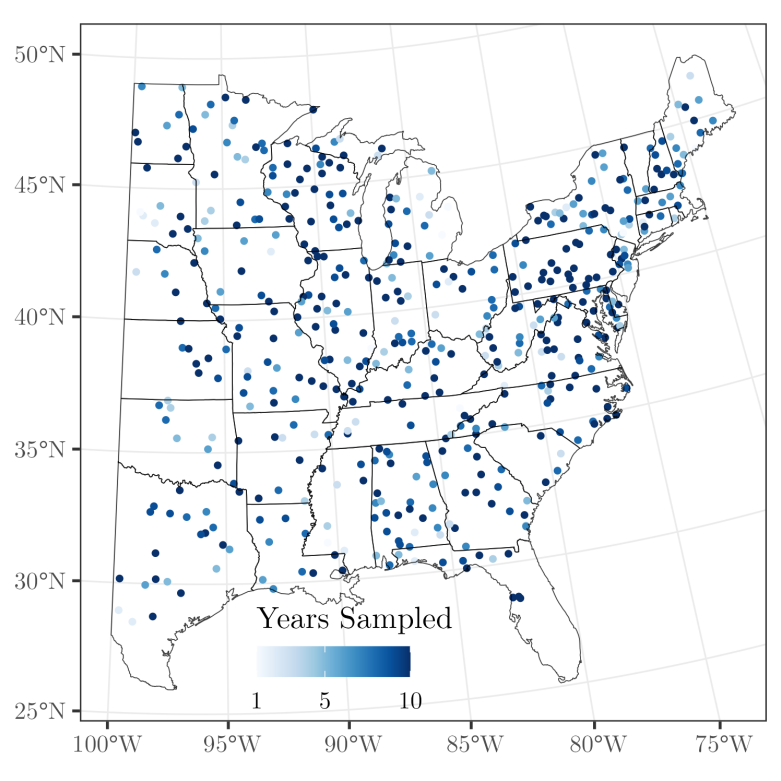


Software

- `spBayes`: univariate Gaussian SVC with full GPs
- `spOccupancy`: univariate Binomial SVC with NNGPs (multivariate on its way)
- `varycoef`: maximum likelihood Gaussian SVCs (Dambon et al. 2021 *Spatial Stats.*)
- `sdmTMB`: penalized likelihood and Bayesian SVC GLMMs (Anderson et al. 2022 *bioRxiv*)

Exercise: 10-year occurrence trend of Wood Thrush

- Data come from USGS North American Breeding Bird Survey
- We desire to account for observational biases in detection of the birds (i.e., false negatives).
- Add on an additional observational layer to our hierarchical model



Exercise: Process model

- Let $z_t(\mathbf{s}_i)$ denote the true presence (1) or absence (0) of the species at site $i = 1, \ldots, 500$ during year $t = 1, \ldots, 10$.

Exercise: Process model

- Let $z_t(\mathbf{s}_i)$ denote the true presence (1) or absence (0) of the species at site $i = 1, \ldots, 500$ during year $t = 1, \ldots, 10$.
- If the bird was detected at a site and year, we know $z_t(\mathbf{s}_i) = 1$. If not, it might be there and we just missed it during the surveys.

Exercise: Process model

- Let $z_t(\mathbf{s}_i)$ denote the true presence (1) or absence (0) of the species at site $i = 1, \ldots, 500$ during year $t = 1, \ldots, 10$.
- If the bird was detected at a site and year, we know $z_t(\mathbf{s}_i) = 1$. If not, it might be there and we just missed it during the surveys.
- We model $z_t(\mathbf{s}_i)$ just as before with a Bernoulli GLM, with a SVC for trend

$$z_t(\mathbf{s}_i) \sim \text{Bernoulli}(\psi_t(\mathbf{s}_i))$$
$$\text{logit}(\psi_t(\mathbf{s}_i)) = \tilde{\beta}_0(\mathbf{s}_i) + \tilde{\beta}_1(\mathbf{s}_i) \cdot \text{YEAR}_t$$

- $\tilde{\beta}_0(\mathbf{s}_i)$ and $\tilde{\beta}_1(\mathbf{s}_i)$ are modeled as independent SVCs with NNGPs

Exercise: Observation model

- Let $y_{t,k}(\mathbf{s}_i)$ denote the observed detection (1) or nondetection (0) of the bird at site i during year t and survey $k = 1, \ldots, 5$.
- We model $y_{t,k}(\mathbf{s}_i)$ conditional on the true presence/absence of the species $z_t(\mathbf{s}_i)$

$$y_{t,k}(\mathbf{s}_i) \mid z_t(\mathbf{s}_i) \sim \text{Bernoulli}(p_{i,t,k} \cdot z_t(\mathbf{s}_i))$$
$$\text{logit}(p_{i,t,k}) = \alpha_{0,t} + \alpha_1 \cdot \text{DAY}_{i,t,k} + \alpha_2 \cdot \text{DAY}^2_{i,t,k}$$

- A key assumption for identifiability is that $z_t(\mathbf{s}_i)$ does not change across the 5 replicate surveys at site i during year t.