### LatticeKrige

### Group member: Fan, Bai, Pammer

• Recall the original model:  $Y = X\beta + Z + E$ 

• If the data is large enough, the calculation of the inverse of  $Cov(Z) = \Sigma(\theta)$  is time-consuming when using mle.

• Low rank method: Z = poly(lon, lat).  $\Sigma(\theta)$  is hard to approximate.

Approximate the covariance function

- $Z_i = \sum_{l=1}^L g_l(s_i)$
- $g(s_i) = \sum_{j=1}^m c_j \phi_j(s_i)$
- $\phi_j(s_i) = \phi(||u_j s_i||/\theta)$



distance



Approximate the covariance function

- $Z_i = \sum_{l=1}^L g_l(s_i)$
- $g(s_i) = \sum_{j=1}^m c_j \phi_j(s_i)$
- $\phi_j(s_i) = \phi(||u_j s_i||/\theta)$





Approximate the covariance function

- $Z_i = \sum_{l=1}^L g_l(s_i)$
- $g(s_i) = \sum_{j=1}^m c_j \phi_j(s_i)$
- $\phi_j(s_i) = \phi(||u_j s_i||/\theta)$





lon

Approximate the covariance function

- $Z_i = \sum_{l=1}^L g_l(s_i)$
- $g(s_i) = \sum_{j=1}^m c_j \phi_j(s_i)$
- $\phi_j(s_i) = \phi(||u_j s_i||/\theta)$





lon

### Simplify the calculation

$$g_{l}(s_{i}) = \sum_{j=1}^{m(l)} c_{j}^{l} \phi_{j,l}(s_{i}) \quad \operatorname{cov}(g_{l}(s), g_{l}(s')) = \sum_{j,k=1}^{m(l)} \rho Q_{j,k}^{-1} \phi_{j,l}(s), \phi_{k,l}(s')$$
  
For each level I:

 $C = (c_1, ..., c_{m(l)}) \sim MN(0, Q^{-1}) \quad \hat{C} = Q^{-1} \phi^T M_{\lambda}^{-1} (Y - X\beta) \quad M_{\lambda}^{-1} = \phi Q^{-1} \phi^T + \lambda W^{-1}$ 

The calculation of  $\hat{C}$  is fast



### **Turning parameters**

NC: grid points

nlevel: level of grid

nu: control of variance for each grid level  $\alpha_j = var(g_l(s_i)), \alpha_j = e^{-2jv}$ 

# Implementation details

- R package: LatticeKrig
- Functions used:

LKrigSetup(): create a list describing the LatticeKrig spatial model. LKinfo <- LKrigSetup( x=, NC = , nlevel = , a.wght = , nu = )

LatticeKrig(): fit the LatticeKrig spatial model to a dataset. fitFinal <- LatticeKrig( x=, y=, LKinfo= LKinfo)

Predict(): spatial prediction.
Y\_hat <- predict(fitFinal, xnew =)</pre>

# Implementation details

- Tuning parameters
- NC

The maximum number of lattice grid points at the coarsest level of resolution. For a example, for a square region, NC=5 results in a 5X5 = 25 lattice

nlevel

Number of levels in multi-resolution. Note that each subsequent level increases the number of basis functions within the spatial domain size by a factor of roughly 4.

• a.wght

For a stationary model, at level k the center point has weight 1 with the 4 nearest neighbors given weight -1/a.wght[k]. In this case a.wght must be greater than 4.

#### • nu

A smoothness parameter that controls relative sizes of alpha. Currently this parameter only makes sense for the 2D rectangular domain.

# Implementation details

• Tuning method

Use observations with g=1 to fit the model. Predict response for g=21 and calculate prediction error. Compare results based on RMSE and fitting time.

- Tuning grid
- NC = 20, 30, 40
- nlevel = 2, 3, 4
- a.wght = 4.1, 6, 8, 10
- nu = 0.1, 0.3, 0.5

### **NC STATE UNIVERSITY**

### RMSE

a.wght

nlevel=2, NC=20 nlevel=2, NC=30 nlevel=2, NC=40 0.1315 0.131 0.13 0.131 0.13 0.129 0.1305 0.13 0.13 U 0.1295 0.129 BU 0.129 전 0.128 BU 0.128 전 0.127 0.129 0.1285 0.127 0.126 0.128 0.1275 0.126 0.125 3 5 10 11 3 5 6 9 10 11 3 5 6 10 11 4 6 7 8 9 4 7 8 4 7 8 9 a.wght a.wght a.wght nlevel=3, NC=20 nlevel=3, NC=30 nlevel=3, NC=40 0.13 0.13 0.129 0.129 0.128 0.129 0.128 망원 0.127 원실 0.126 0.128 U.128 0.126 0.127 0.125 0.125 0.126 0.124 0.124 3 5 6 7 9 10 11 3 5 9 10 11 10 11 4 8 4 3 5 6 7 9 4 8 a.wght a.wght a.wght nlevel=4, NC=20 nlevel=4, NC=30 nlevel=4, NC=40 0.129 0.129 0.128 0.127 0.128 0.128 망 0.127 일 0.126 망 0.126 전 0.125 BS 0.127 0.126 0.124 0.125 0.124 0.125 0.123 3 5 6 7 8 9 10 11 5 7 9 10 11 5 6 10 11 4 3 4 6 8 3 4 7 8 9 a.wght

Speaker: Bai

a.wght

### **NC STATE UNIVERSITY**

Fitting time



Speaker: Bai

#### Speaker: Jacob



### Results

- MLE is Cubic
- Lattice Kriging shows relatively consistent linearity
- When there are more grid points and mor n levels it grows at a much faster rate. (10X)

#### NC = 40, NLevel = 4, aweight = 6, nu = .3



Number of Training Sets Used

0 1000 800 Time Took 009 0 400 0 200 0 0 0 5 2 3 1

NC = 20, NLevel = 4, aweight = 6, nu = .3



MLE estimator

Number of training sets used

#### **NC STATE UNIVERSITY**

Speaker: Jacob

NC = 20, NLevel = 3, aweight = 6, nu = .3 0 0.125 0 0 0 0 0.120 RMSE  $\circ$ 0.115 0 0 0 0 20 5 10 15 Number of Training Sets Used NC = 40, NLevel = 3, aweight = 6, nu = .30.125 0 0.120 0 0 0 0 0.115 RMSE 0.110 0

### Results

- Lattice has worse RMSE values (not by much)
- The use of 20 training sets in kriging in is more accurate than just 5 for MLE





#### NC = 20, NLevel = 4, aweight = 6, nu = .3



NC = 40, NLevel = 4, aweight = 6, nu = .3



10 Number of Training Sets Used

5

0

0

15

0

20

0.105

Number of Training Sets Used

Speaker: Jacob

## Results

- When Lattice Kriging splits up the data into grids, it enables the computer to process far less computations.
- It does not compromise accuracy too much
- You can have many gridlines (Adding time and accuracy)