LatticeKrige

Group member: Fan, Bai, Pammer

Speaker: Fan

Introduction

- 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.

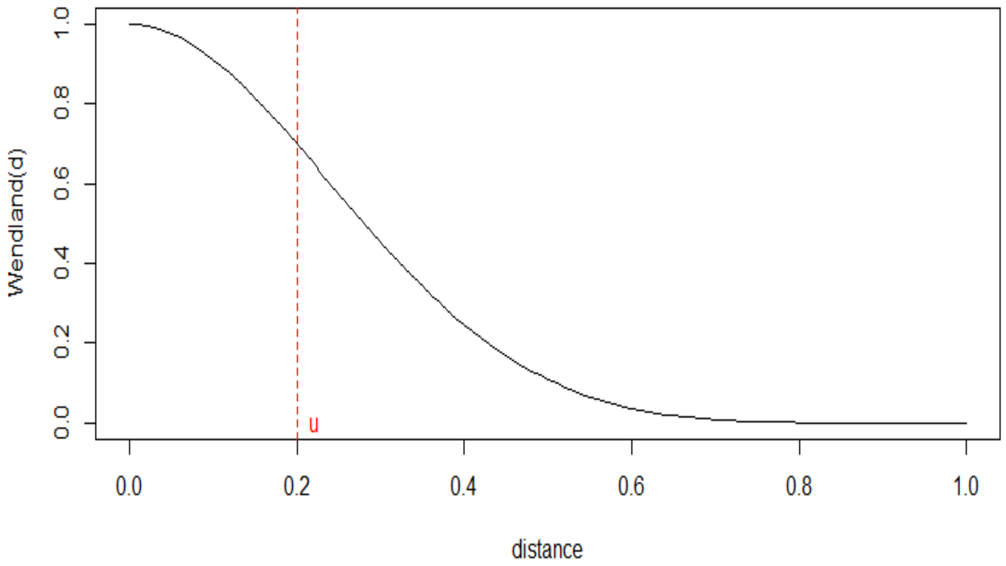
- Low rank method: $Z = poly(lon, lat)$. $\Sigma(\theta)$ is hard to approximate.

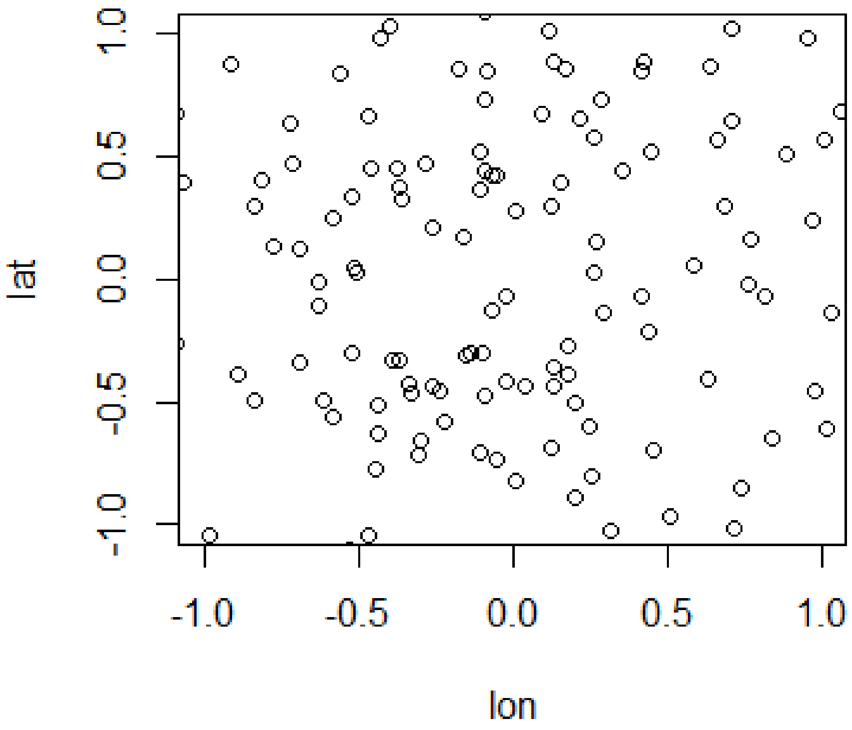
Speaker: Fan

Introduction

Approximate the covariance function

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



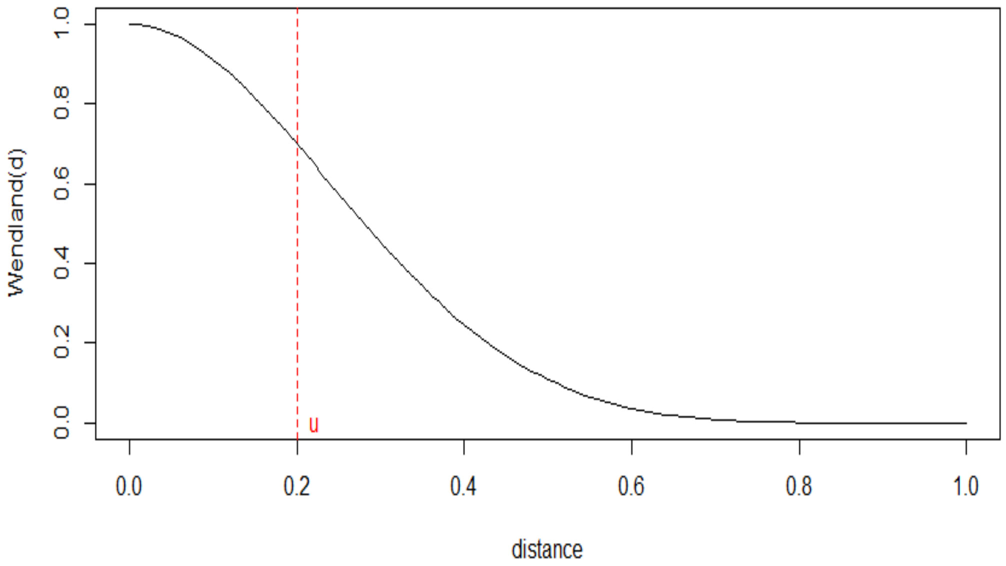


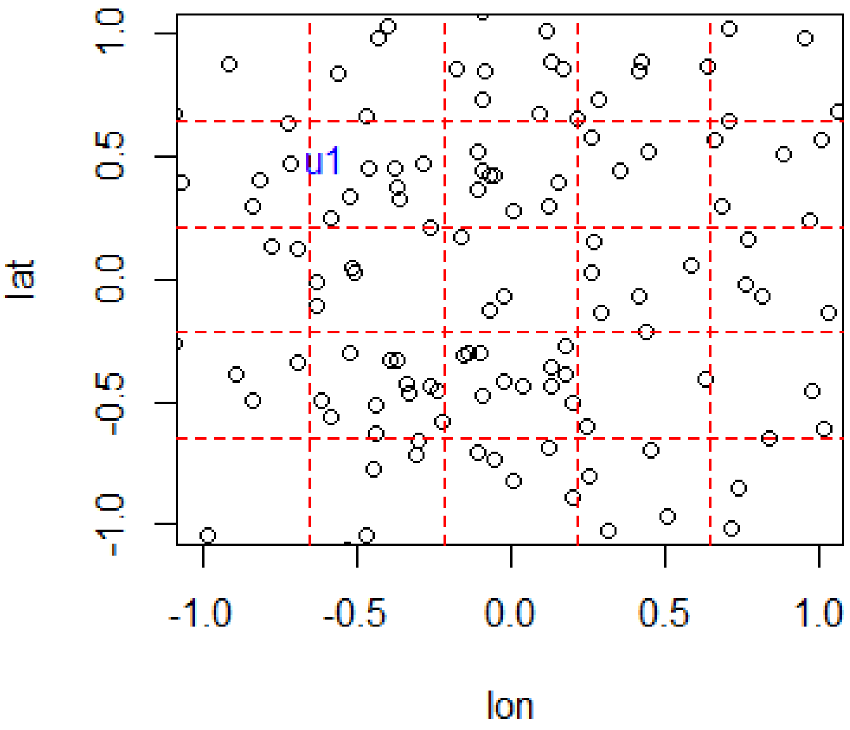
Speaker: Fan

Introduction

Approximate the covariance function

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



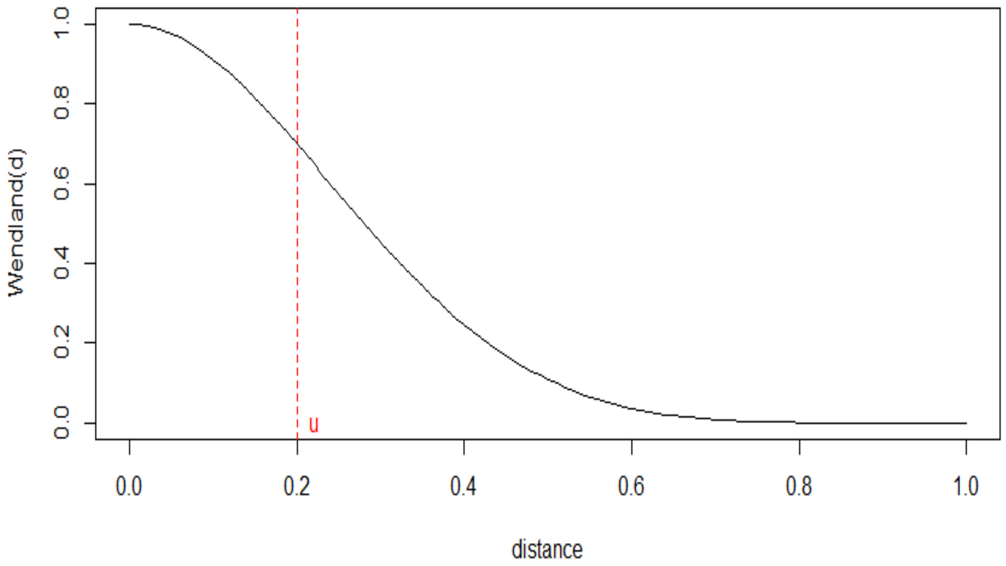


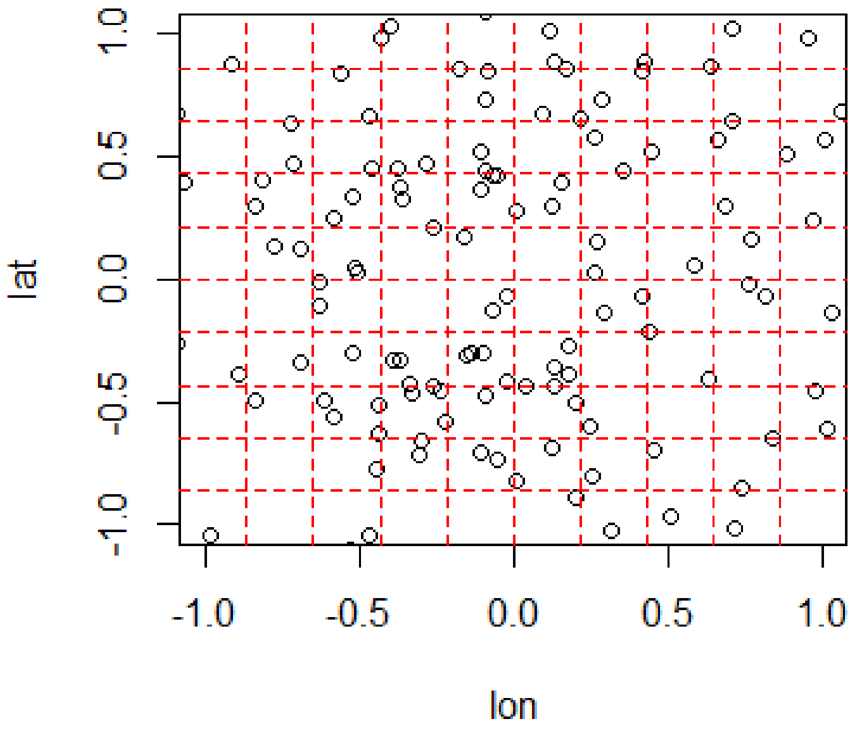
Speaker: Fan

Introduction

Approximate the covariance function

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



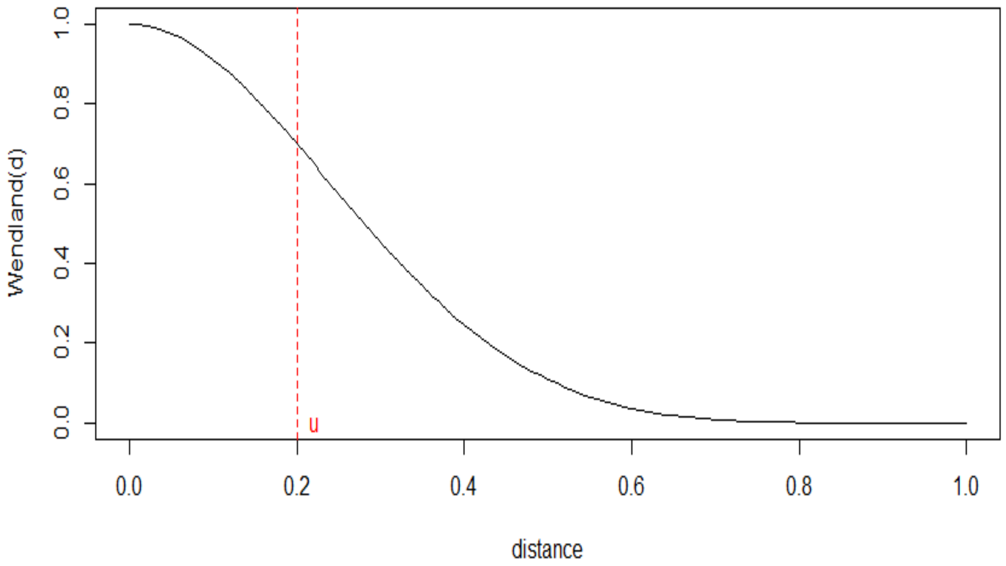


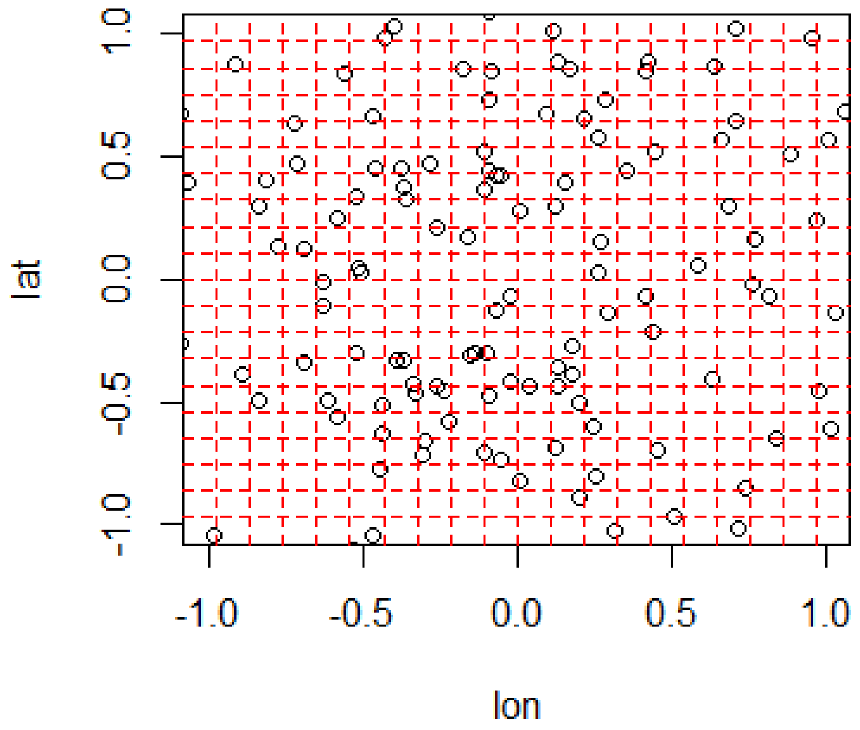
Speaker: Fan

Introduction

Approximate the covariance function

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





Speaker: Fan

Introduction

Simplify the calculation

$$g_l(s_i) = \sum_{j=1}^{m(l)} c_j^l \phi_{j,l}(s_i) \quad \mathrm{cov}(g_l(s), g_l(s')) = \sum_{j,k=1}^{m(l)} \rho Q_{j,k}^{-1} \phi_{j,l}(s), \phi_{k,l}\left(s'\right)$$

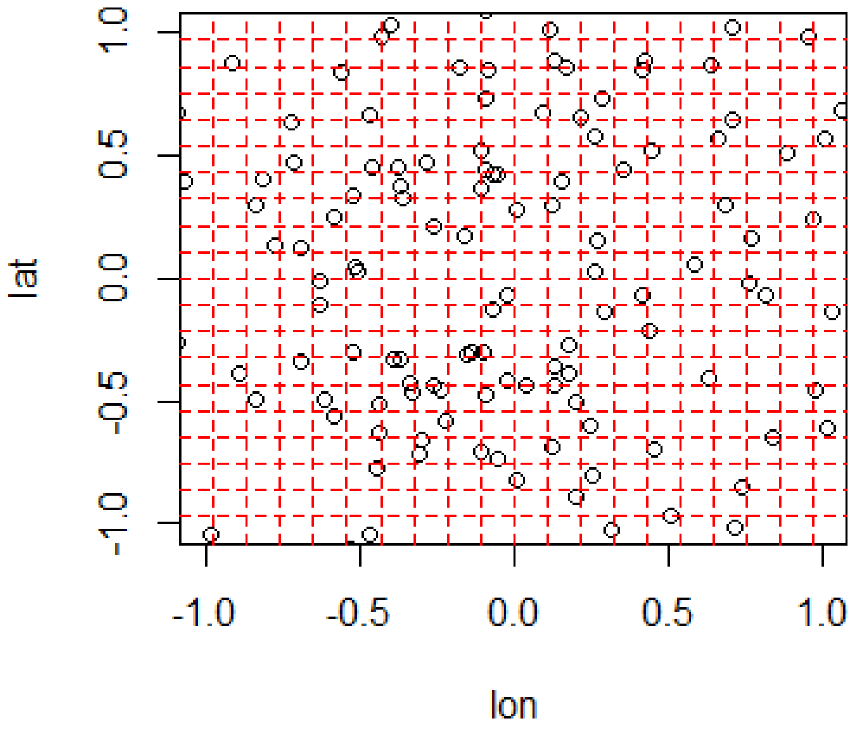
For each level l:

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

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

Speaker: Fan

Introduction



Turning parameters

NC: grid points

nlevel: level of grid

nu: control of variance for each grid level

$\alpha_j = var(g_l(s_i)), \alpha_j = e^{-2j\nu}$

Speaker: Bai

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 =)
```

Speaker: Bai

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.

Speaker: Bai

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

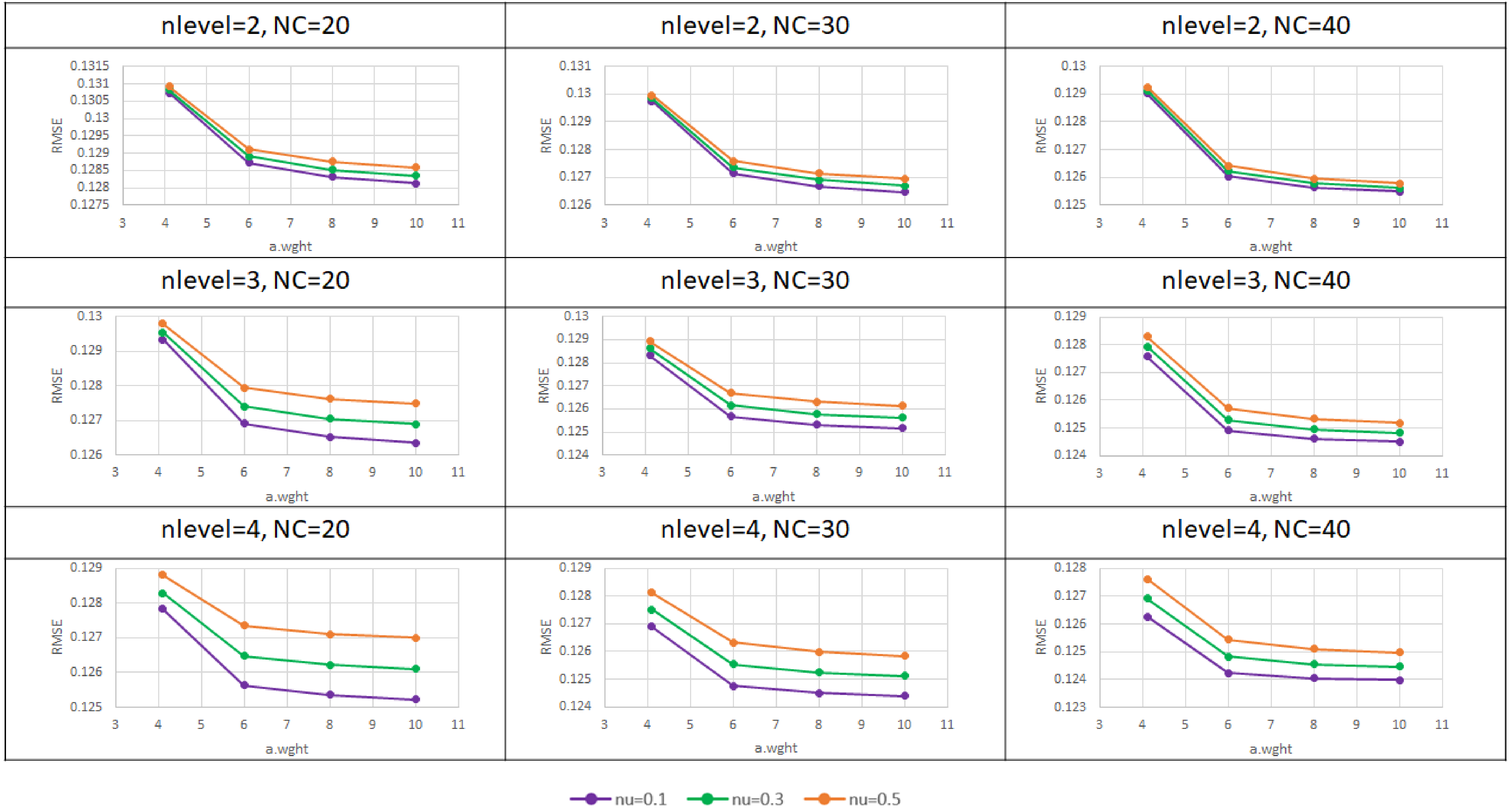
nlevel = 2, 3, 4

a.wght = 4.1, 6, 8, 10

nu = 0.1, 0.3, 0.5

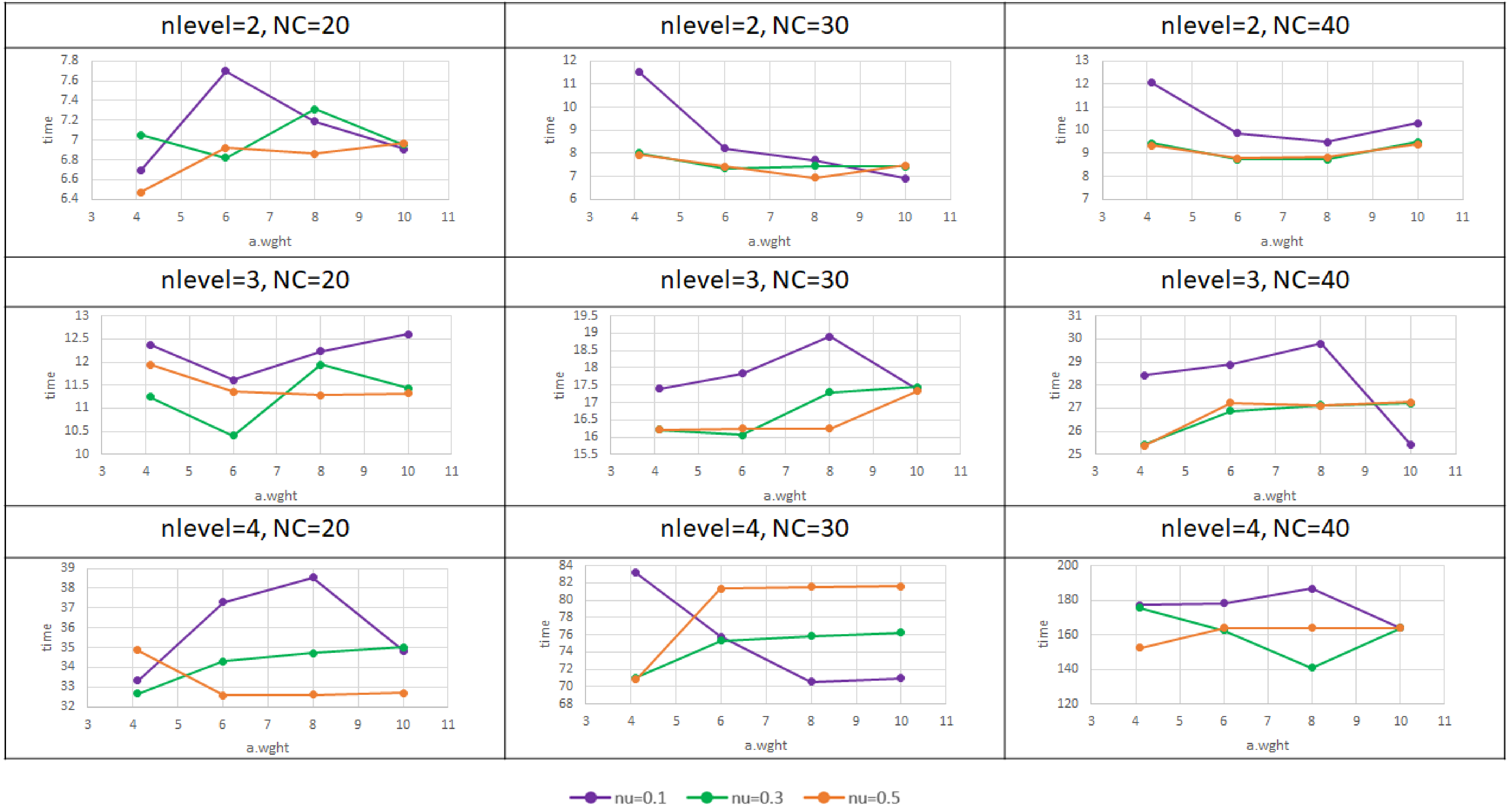
RMSE

Speaker: Bai



Fitting time

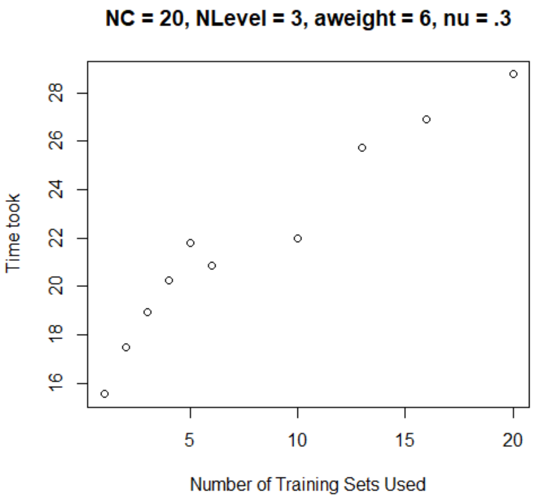
Speaker: Bai

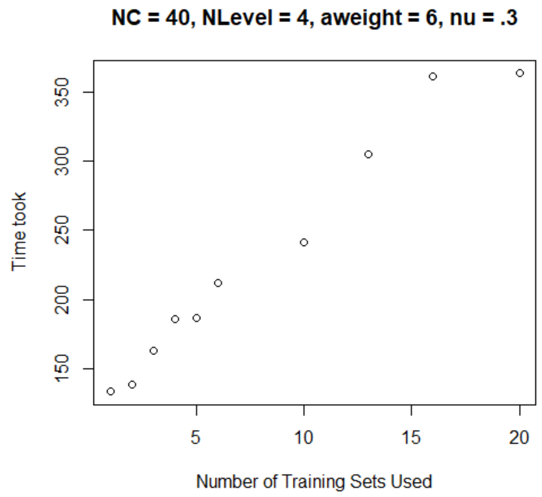


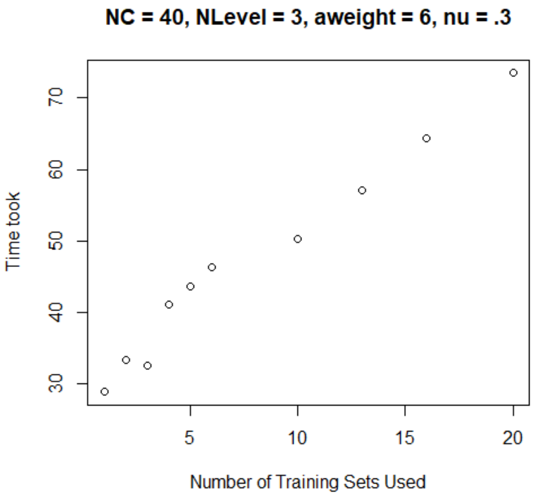
Speaker: Jacob

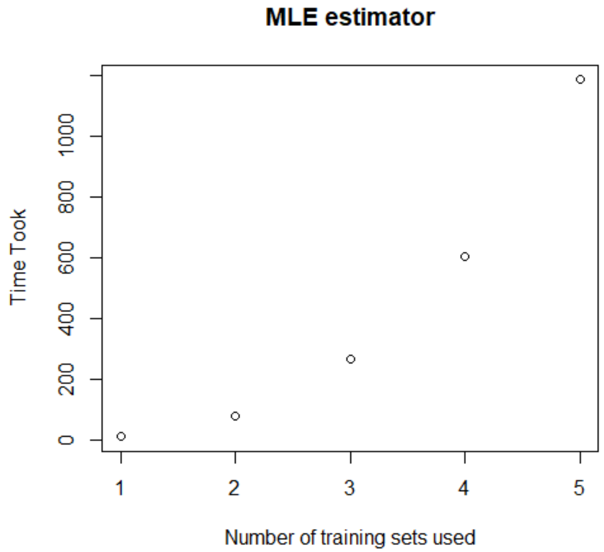
Results

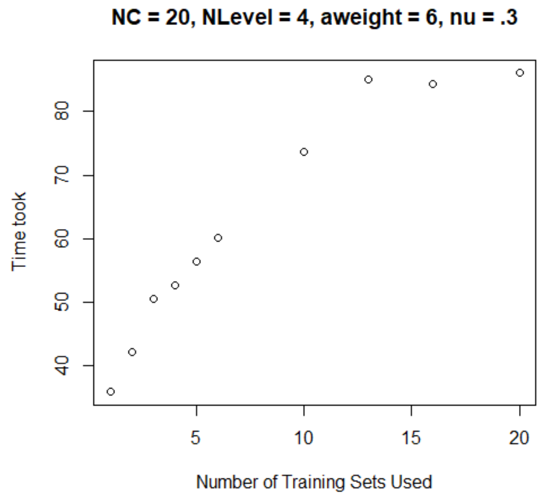
- MLE is Cubic
- Lattice Kriging shows relatively consistent linearity
- When there are more grid points and mor n levels it grows at a much faster rate. (10X)







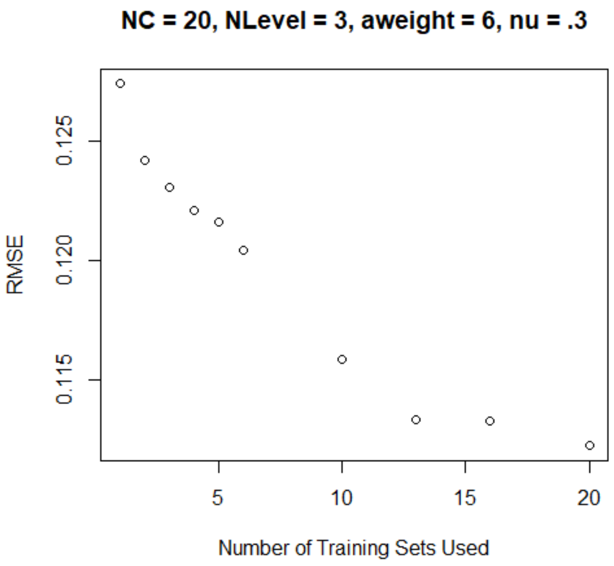


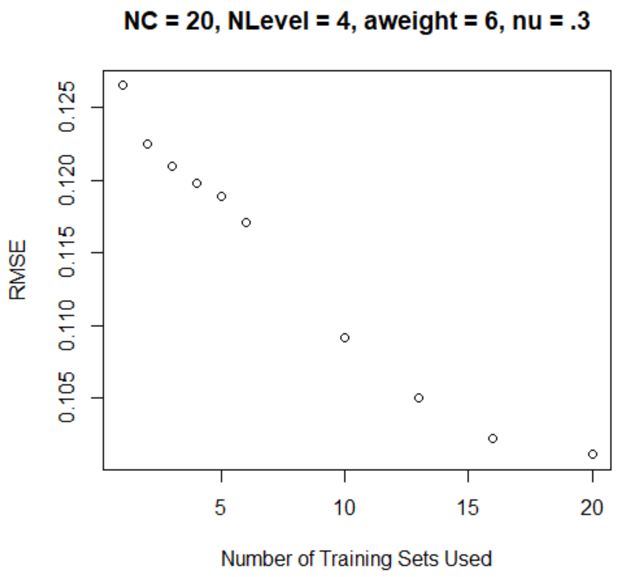


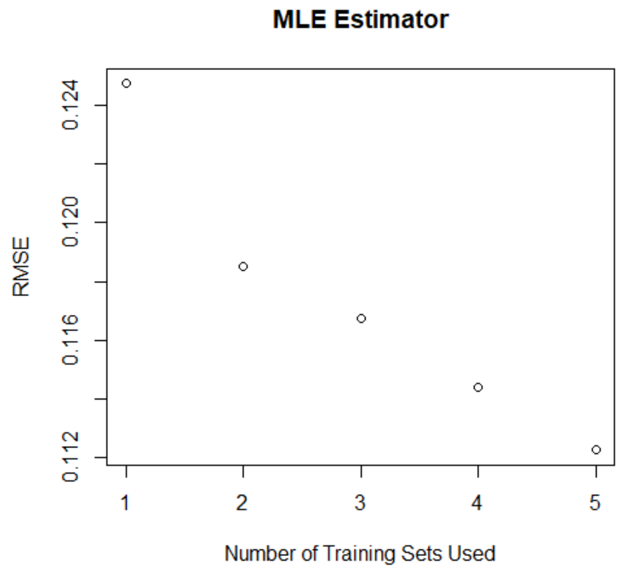
Speaker: Jacob

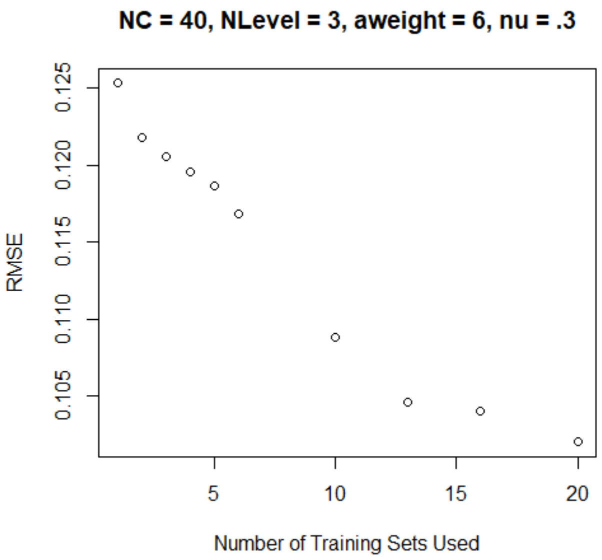
Results

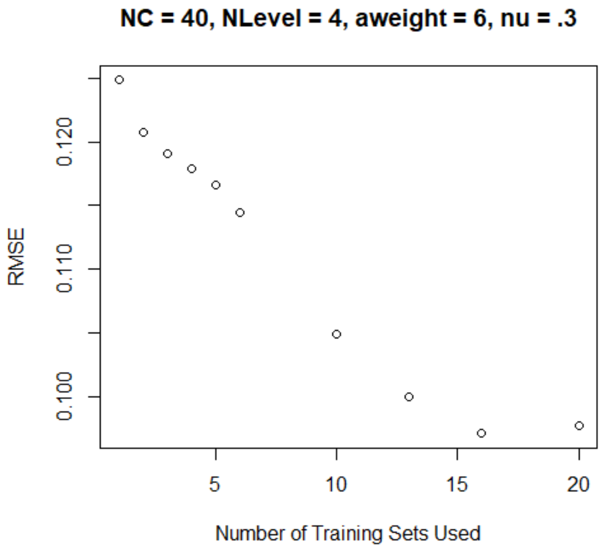
- Lattice has worse RMSE values (not by much)
- The use of 20 training sets in kriging in is more accurate than just 5 for MLE











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)