

Predictive Processes

ST 533 Midterm 2 Project

Johnny, Emine, Hunter

Predictive Process model in layman's

Say you have to specify a model at a million points.

But you can't do a million points exactly, because that's too cumbersome.

So instead you do it at exactly 100 points. And then instead of specifying the full model on a million points, I'm just going to connect the 100 points I specified the model for with the other million–100 points using kriging equations.

Therefore the model goes from trying to build a model for the whole dataset, to a great model for a subset of the data and then connected with the kriging equations.

Description of Process

- Predictive Process is a type of Fixed Rank Kriging
- Similar to our previous spatial modeling
- Hierarchical model
- “Parent” spatial random effects process occurs at the $N \times N$ observations and it gets “propagated” to the $K \times K$ knots
- This strategy improves process time due to the the estimation of the latent Gaussian process at $K \ll N$ knots
- Instead of factoring the $N \times N$ covariance matrix, we factorize the $K \times K$ covariance matrix at the knot locations

Formula

- Consider the spatial random effects model: $Y(\mathbf{s}) = \mathbf{x}(\mathbf{s})' \boldsymbol{\beta} + w(\mathbf{s}) + \epsilon(\mathbf{s})$
- Assume that $w(\mathbf{s})$ follow a mean zero Gaussian process with covariance function $\mathbb{C}(\mathbf{s}, \mathbf{s}') = \sigma_w^2 \rho(\mathbf{s}, \mathbf{s}')$ where $\rho(\cdot, \cdot)$ is a positive-definite correlation function
- Let $\mathbf{s}_1^*, \dots, \mathbf{s}_K^*$ denote the set of knot locations in the spatial domain
- Now, $\mathbf{w}^* = (w(\mathbf{s}_1^*), \dots, w(\mathbf{s}_K^*))' \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{w^*})$ where $\boldsymbol{\Sigma}_{w^*}$ is a $K \times K$ covariance matrix with ij th element $\mathbb{C}(\mathbf{s}_i^*, \mathbf{s}_j^*)$
- Now we replace $w(\mathbf{s})$ with $\tilde{w}(\mathbf{s}) = \mathbb{C}'(\mathbf{s}, \mathbf{s}^*) \boldsymbol{\Sigma}_{w^*}^{-1} \mathbf{w}^*$ where $\mathbb{C}(\mathbf{s}, \mathbf{s}^*) = (\mathbb{C}(\mathbf{s}, \mathbf{s}_1^*), \dots, \mathbb{C}(\mathbf{s}, \mathbf{s}_K^*))'$

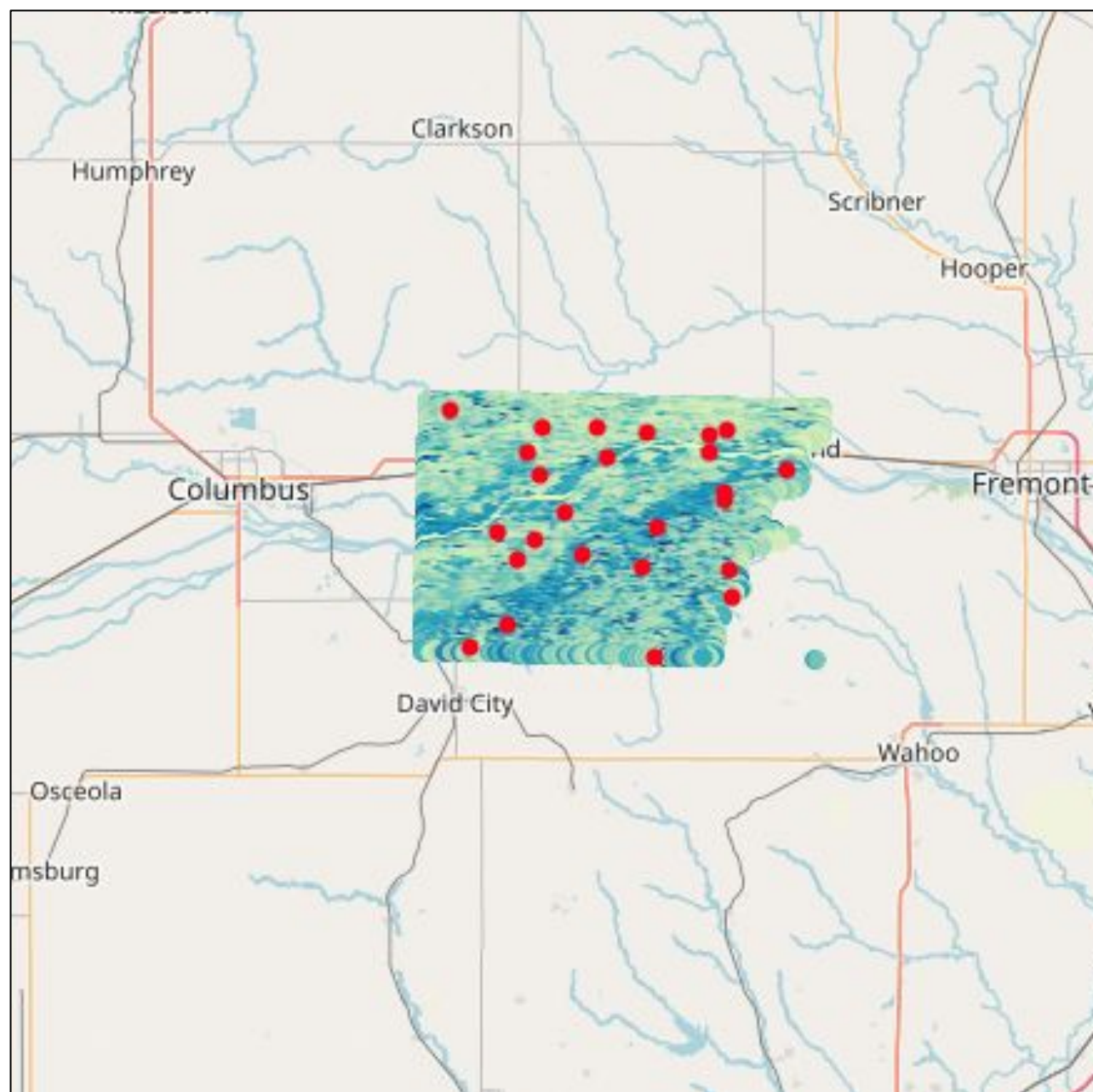
Formula: Underestimated Variance

- A shortcoming of the original formulation of the predictive process is that it induces a positive bias in the non-spatial error term of the models.
- A modified predictive process is proposed to address this problem.
- To counteract this: $Y(\mathbf{s}) = \mu(\mathbf{s}) + \tilde{w}(\mathbf{s}) + \xi(\mathbf{s}) + \varepsilon(\mathbf{s})$ where $\xi(\mathbf{s})$ is a fine-scale variation process

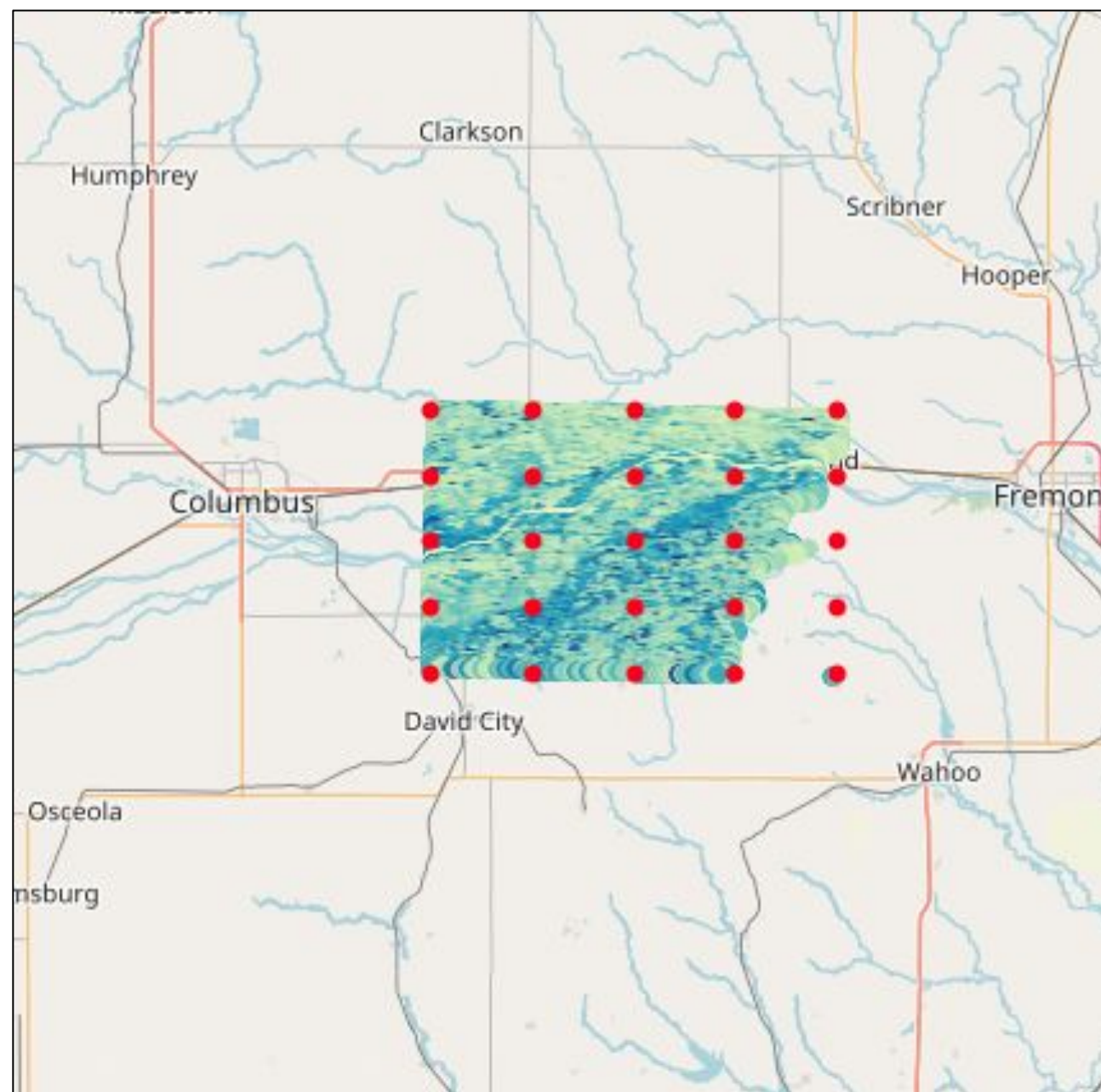


Knot Locations for this data

Study Area with 25 random knots



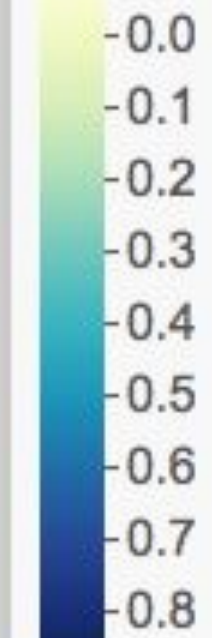
Study Area with 5x5 knots



Legend

● Knot Locations

LandSAT Index Values



How to Apply Predictive Process

- We will use the `spLM()` function from `{spBayes}` to train our Predictive Process model.
- The `spLM()` function can implement predictive process models by specifying the `knots` command.
- Some computation considerations:
 - The greater the size of data the longer the model will take to fit to an exponential degree, which must be considered when partitioning the dataset.
 - The more MCMC sample iterations, the longer the model will run. It's best to start with a smaller sample of iterations, then increase if convergence is not reached.

How to specify knots

- Knot design:
 - 1.) grid/random sample of locations spanning the data extent
 - 2.) space-covering design that optimizes distance (Royle & Nychka, 1998)
 - 3.) locations are based on minimizing predictive variance (Finley et al., 2009)
- Rules of thumb for the number of knots:

There is little difference in prediction between the different knot designs

The number of knots has the greatest impact on parameter estimates and prediction.

The selection of knots should be based on an investigation of prediction sensitivity with knot intensity, with consideration of computational efficiency.

How to specify knots

- Specifying knots in {spBayes}:

1.) a vector of rows and columns specifying a grid

Example 1: `knots <- c(5,5,0) # a 5x5 grid spanning the outter-most extent of the data`

Example 2: `knots <- c(10,10) # a 10x10 grid of 100 locations spanning across the data`

2.) a 2 column matrix of locations

Example 3: `knots <- as.matrix(knot_locations[,1:2]) # manually delineated locations`

- Selecting an unmodified/modified Predictive Process:

Unmodified PP: `modified.pp = F`

Modified PP (default): `modified.pp = T`

Setting up the Predictive Process in R

Import the package

```
library(spBayes)
```

Set up the model using our data, here represented as the variable `dataset`

```
s <- as.matrix(dataset[,1:2]) # locations
p <- 6 # the number of covariates, including intercept
n.samples <- 25000 # total number of MCMC samples
burn <- 5000 # burn the first 5000 sample iterations
maxd <- max(dist(s)) # max distance between data locations
knots <- c(5,5,0) # 5x5 grid of locations for w(s)

starting <- list("phi"=1/(0.05*maxd),
               "sigma.sq"=0.5*var(dataset$Y),
               "tau.sq"=0.5*var(dataset$Y))
tuning <- list("phi"=1,
              "sigma.sq"=0.05*var(dataset$Y),
              "tau.sq"=0.05*var(dataset$Y))
priors <- list("beta.Norm"=list(rep(0,p), 100*diag(p)),
              "phi.Unif"=c(1/(2*maxd), 1/(0.01*maxd)),
              "sigma.sq.IG"=c(2, 1),
              "tau.sq.IG"=c(2, 1))
```

Starting values for estimating model parameters

Variances for the MCMC sampler (this is not needed for amcmc)

Uninformative priors capturing the range of parameter values

```

for(k in 1:20){
  # Create training dataset for this loop
  train <- dataset[g>0 & g<=k,] %>% na.omit()
  test <- dataset[g==21,] %>% na.omit()

  # Fit the linear model using training dataset
  fit <- spLM(Y ~ long + lat + long2 + lat2 +longlat,
             data=train,
             coords=as.matrix(train[,1:2]),
             knots=knots,
             starting=starting,
             tuning=tuning,
             priors=priors,
             cov.model="exponential",
             modified.pp=T, # Modified Predictive Process
             n.samples=n.samples,
             verbose=F)

  # Save model output
  name_model <- paste("fit_", k, sep="")
  assign(name_model,fit)

  # Predict on independent dataset
  pred <- spPredict(fit,
                   pred.coords=as.matrix(test[,1:2]),
                   pred.covars=as.matrix(cbind(1,test[,-3])),
                   thin=10,
                   start=burn,
                   verbose=F)

  # Save predictions
  name_pred <- paste("pred_",k, sep="")
  assign(name_pred, pred$p.y.predictive.samples)
}

```

Data partitioning

Model building

Model exporting

Model application

Prediction exporting

Results

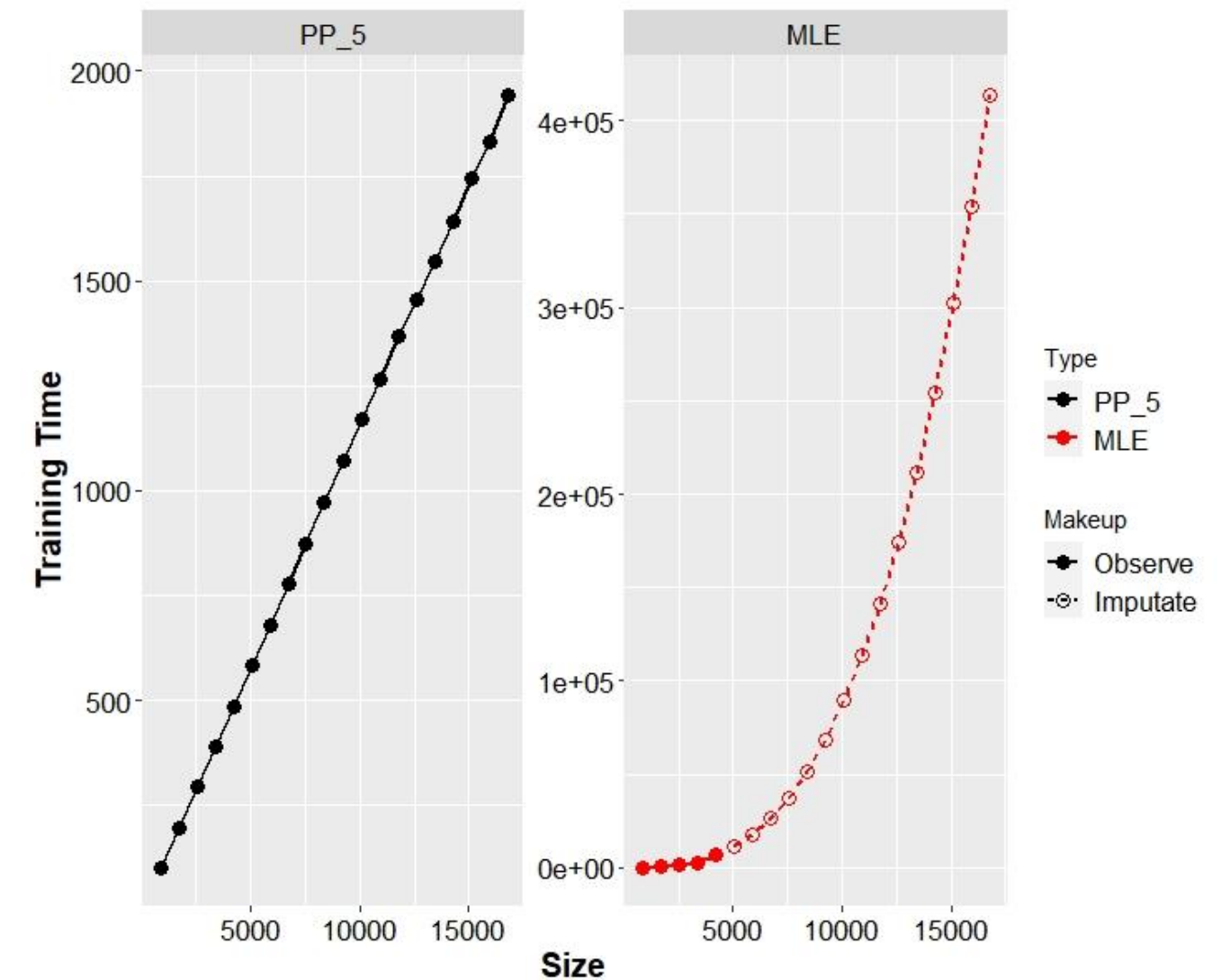
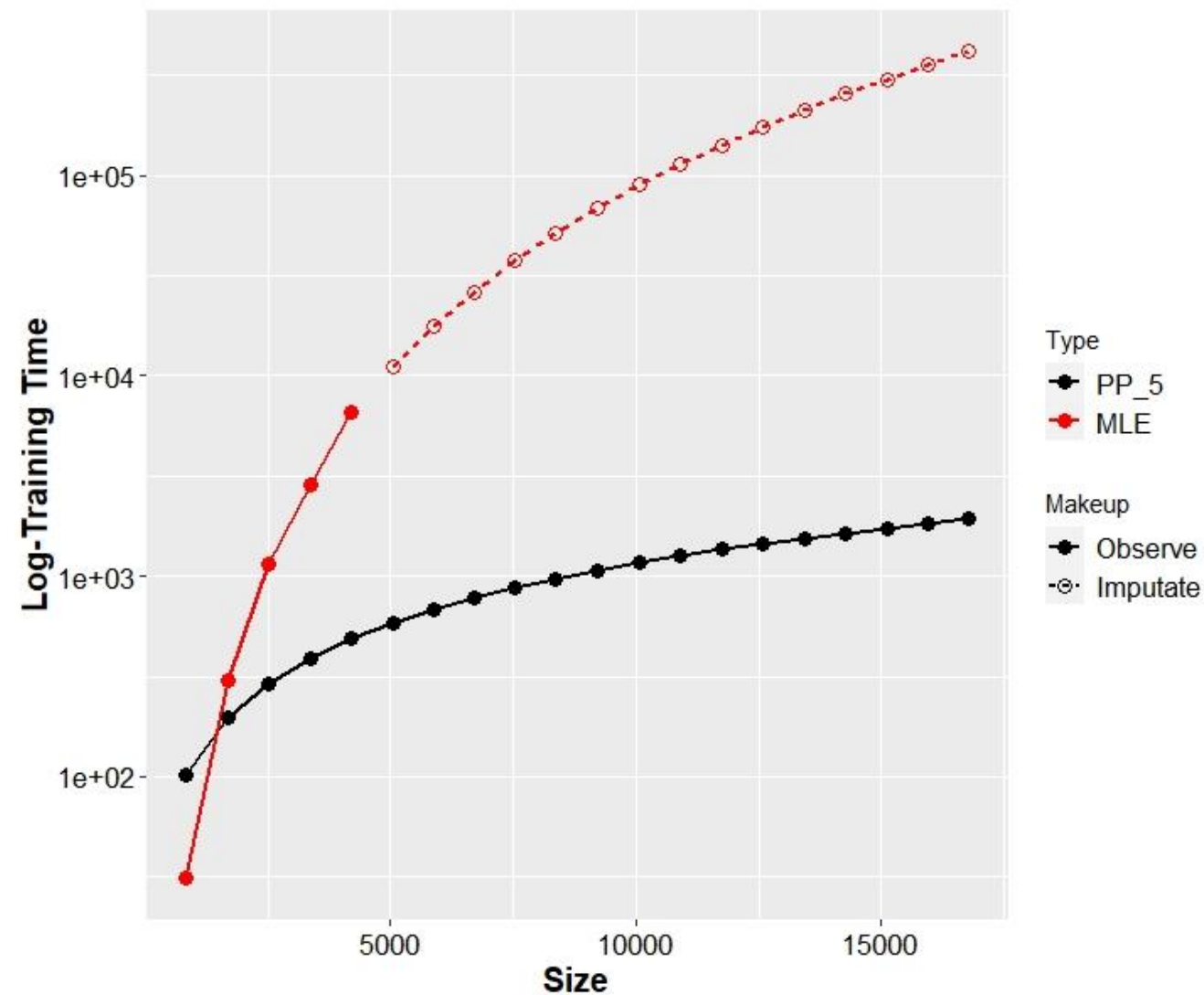
All experiments are coded in RStudio (RStudio Team, 2020) using R (R Core Team, 2020) on a Windows 10 PC with an I5-9600K processor(6 Cores @3.7GHz) and 32GB of RAM.

Considering of the reproducibility of the experiment, we use the seed of 919 and separate the original data into 21 groups (G1 - G21), where G1 to G20 are mainly used for training models with various sample sizes, and G21 serves as the “public” testing set for performance comparison. We will calculate performance metrics as following:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2; \quad \text{COR} = \text{corr}(\mathbf{Y}, \hat{\mathbf{Y}}).$$

Time on the Sample Size

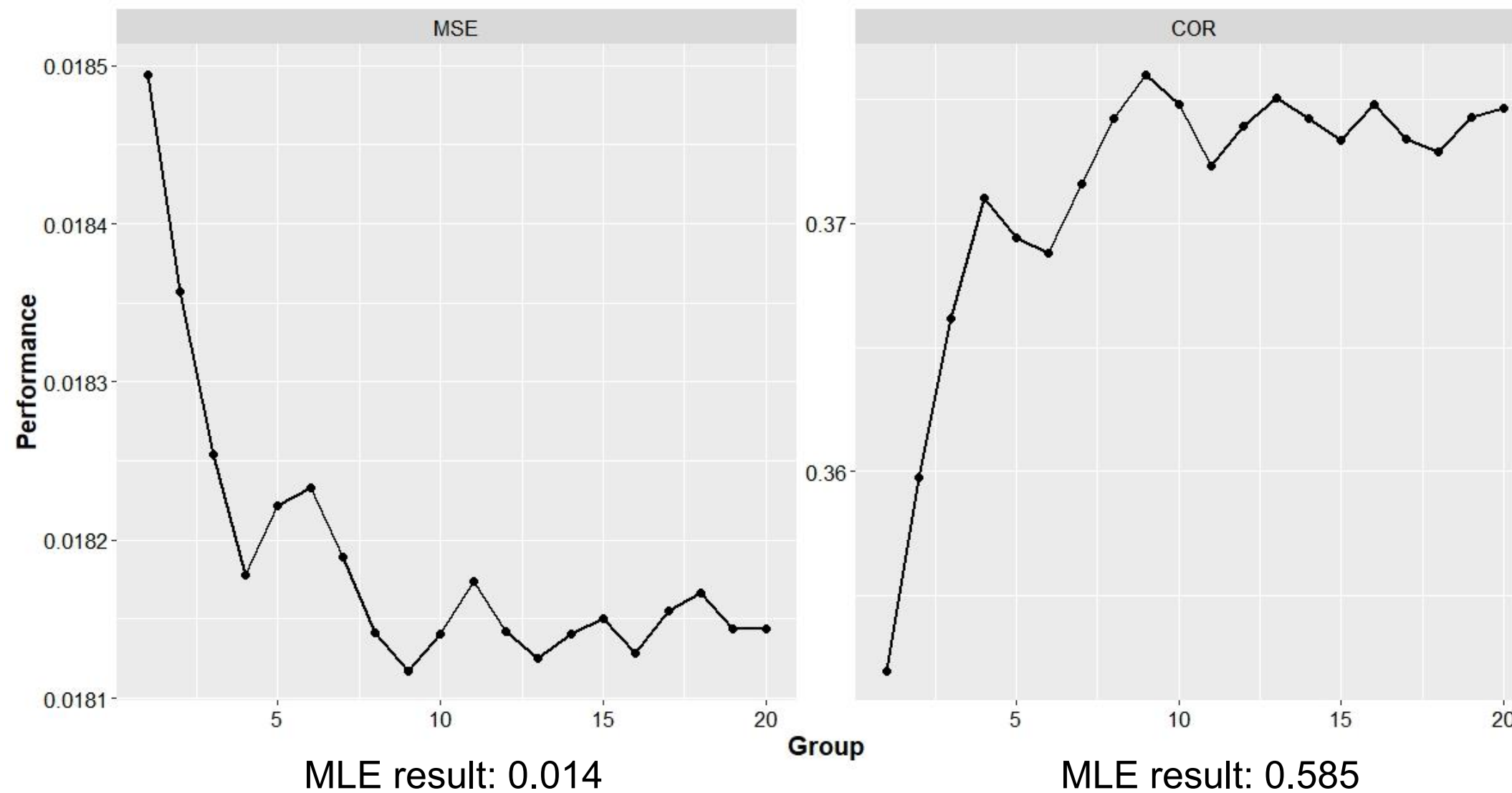
Experiment Settings: 20 steps in total; $G_1 - G_i$ ($i = 1, 2, \dots, 20$) are used for training step i ; G_{21} is the testing set; The number of knots in PP = 5×5 .



Result: The training time of Predictive Process methods linearly increased, $O(N)$, with the number of the observations, while MLE is an $O(N^3)$ process.

Performance on the Sample Size

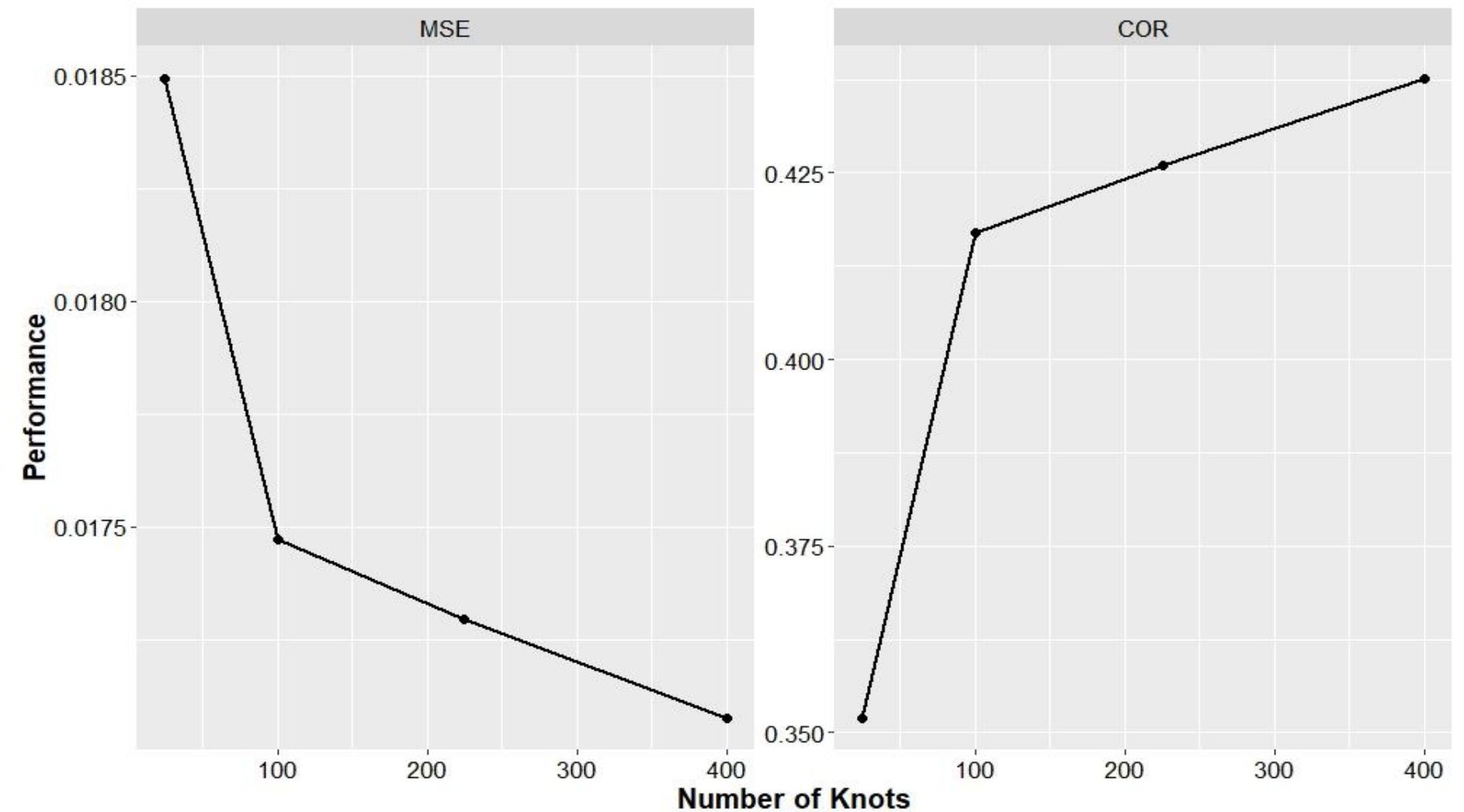
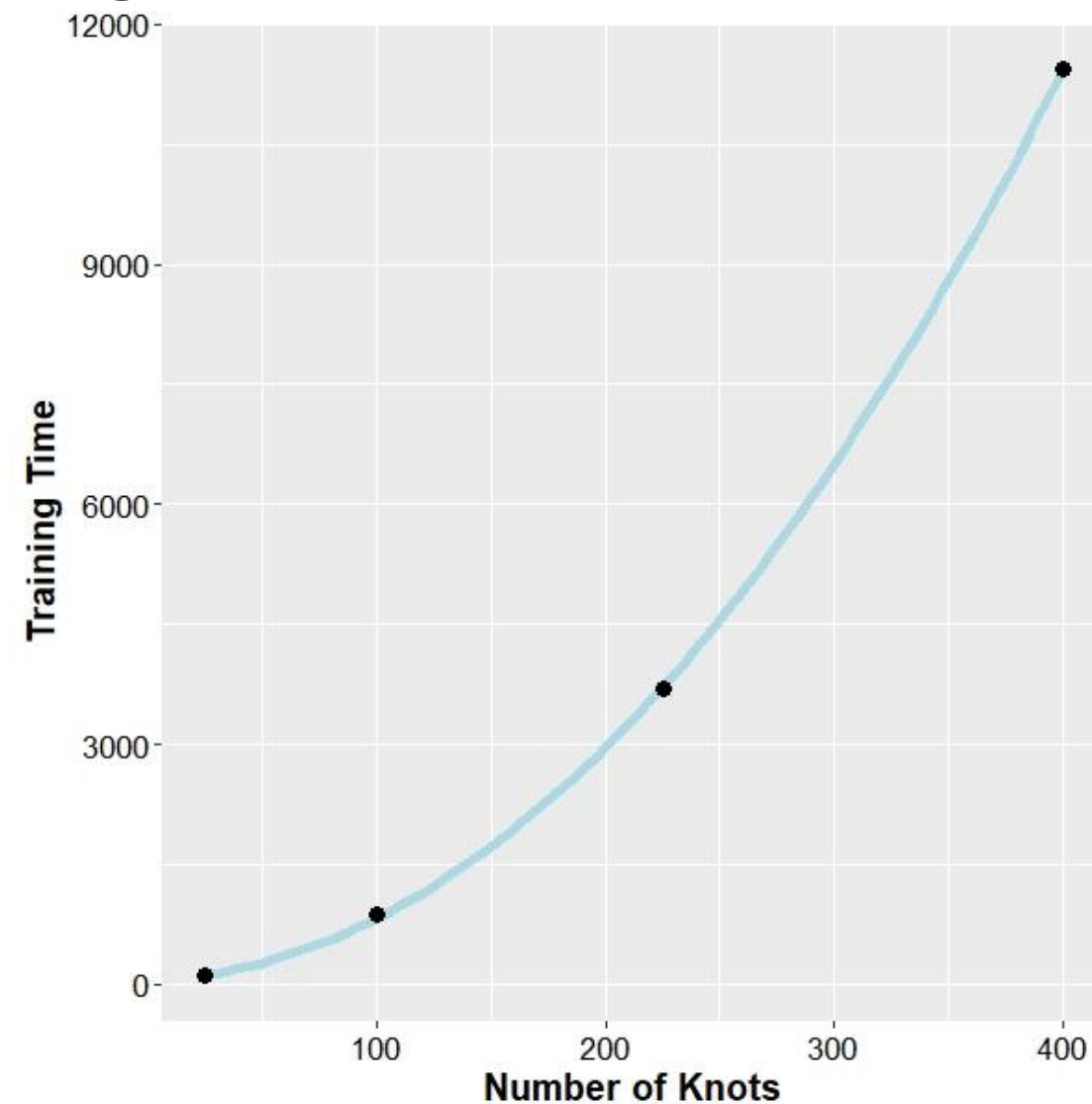
Experiment Settings: 20 steps in total; $G_1 - G_i$ ($i = 1, 2, \dots, 20$) are used for training step i ; G_{21} is the testing set; The number of knots in PP = 5×5 .



Result: The Predictive Process performs worse than the MLE, but may get better when we have a larger data set, and we may get the “maximum” performance in this setting in about a half of the total data.

Results on the Number of Knots

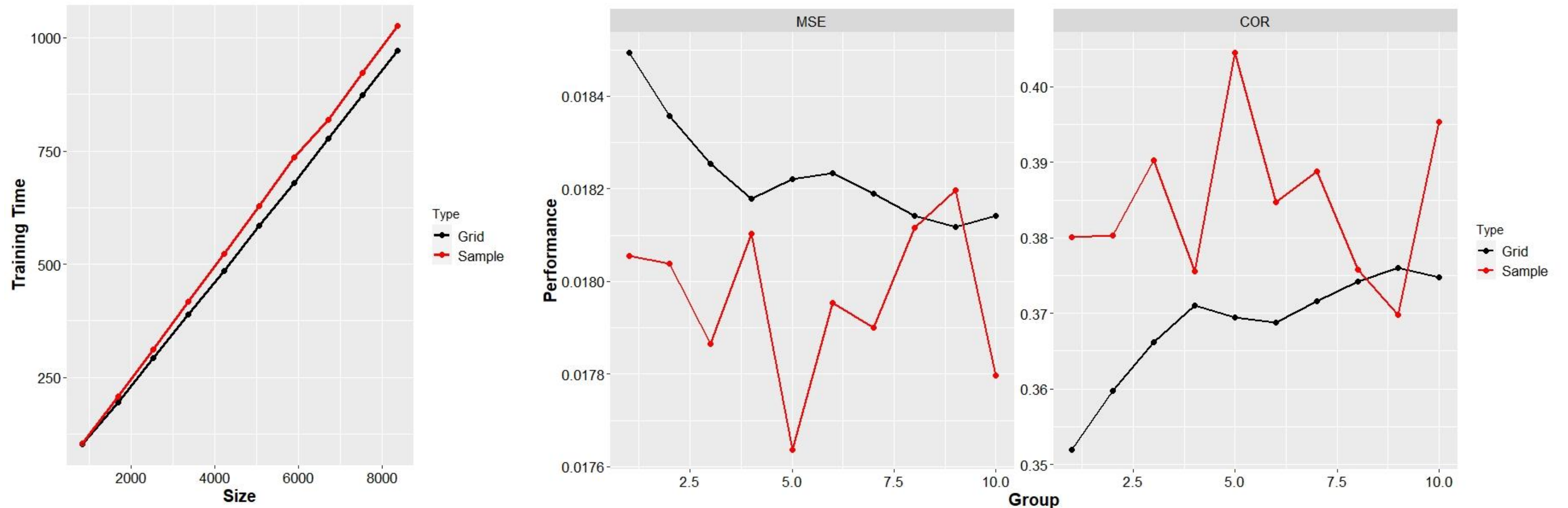
Experiment Settings: 4 steps in total; G1 are used for training; G21 is the testing set; The number of knots in PP = c(5x5, 10x10, 15x15, 20x20).



Result: The training time of Predictive Process methods quadratic increased with the number of the knots, with the increasing performance in MSE & COR.

Results on different knot designs

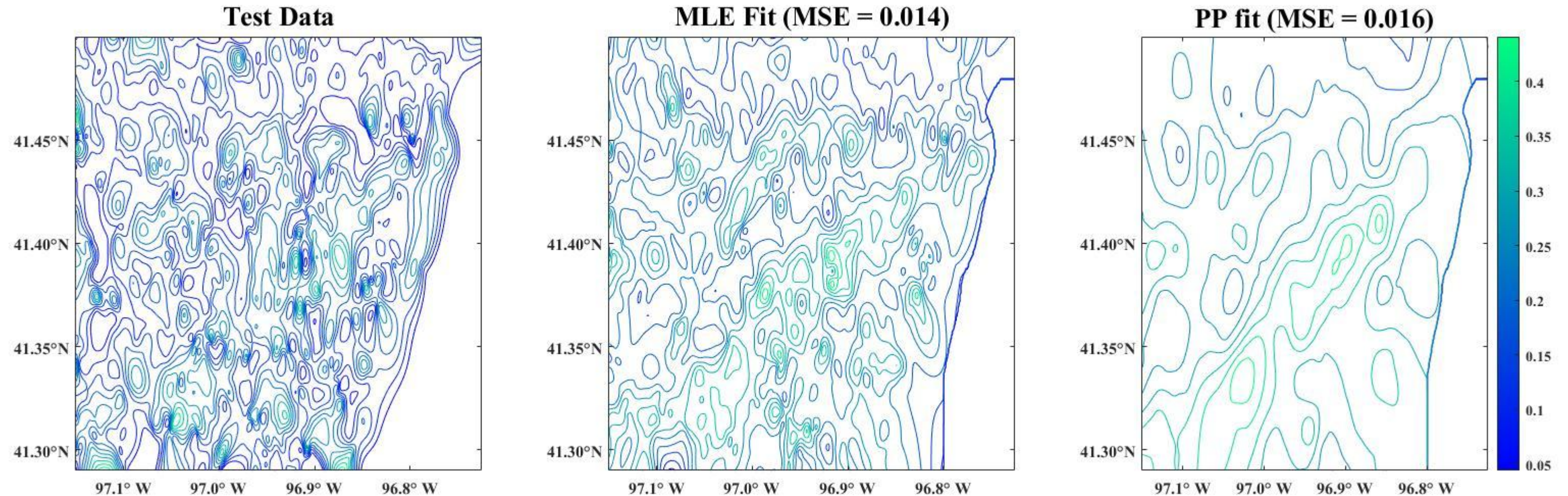
Experiment Settings: 10 steps in total; G1-G_i (i=1,2,...,10) are used for training set; G₂₁ is the testing set; The number of knots in PP = 5x5 / 25.



Result: The training time of two knot designs are pretty similar, but the performance of Sample knots is better (while much bumpy).

Prediction of Predictive Process

Experiment Settings: G1-G5 are used for training set; G21 is the testing set;
The number of knots in PP = 150.



Result: The Predictive Process model finishes in 10587s, and gets a MSE of 0.016. The PP fit is more smooth than the MLE fit.

References

- Finley A., Sang H., Banerjee S., Gelfand A. (2009). “Improving the performance of predictive process modeling for large datasets.” *Computational Statistics & Data Analysis*, 53(8), 2873–2884.
- Finley A., Banerjee S., Gelfand A. (2015). “spBayes for large univariate and multivariate point-referenced spatio-temporal data models.” *Journal of Statistical Software*, 63(13), 1–28.
- Royle J., Nychka D. (1998). “An algorithm for the construction of spatial coverage designs with implementation in SPLUS.” *Computers & Geosciences*, 24(5), 479–488.
- RStudio Team (2020). RStudio: Integrated Development for R. RStudio, PBC, Boston, MA URL <http://www.rstudio.com/>.
- R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.