

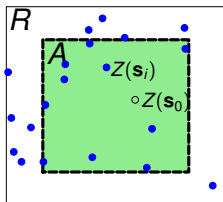
Block Prediction

Jay Ver Hoef

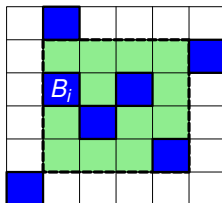
National Marine Mammal Lab
NOAA Fisheries
International Arctic Research Center
Fairbanks, Alaska, USA

Introduction

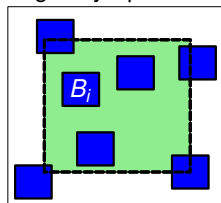
1) Block Kriging



2) Block Prediction for Finite Populations on a Grid

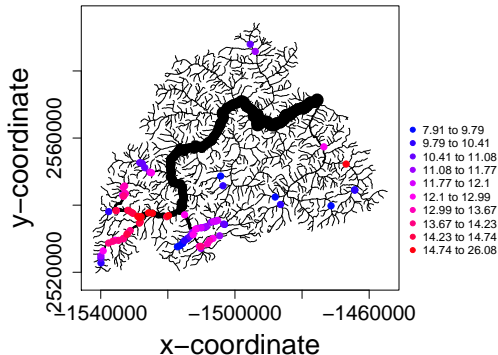


3) Block Prediction for Finite Populations Irregularly Spaced

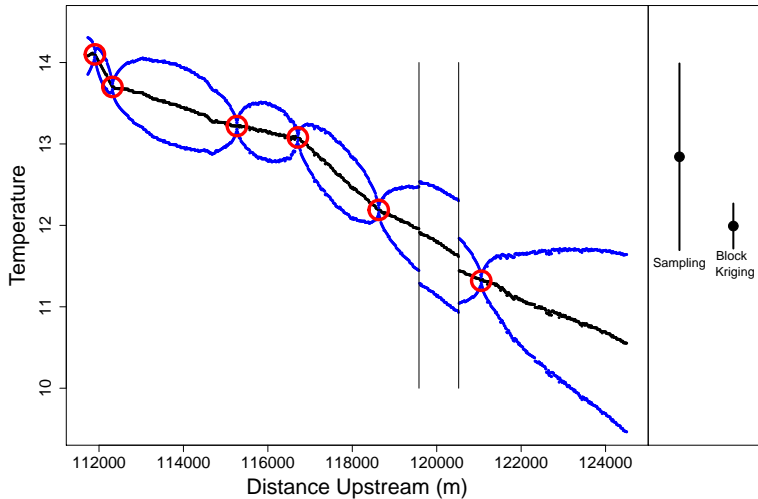


Block Prediction on a Stream Network

Peterson, E.E., Ver Hoef, J.M., Isaak, D.J., Falke, J.A., Fortin, M-J, Jordan, C., McNyset, K., Monestiez, P., Ruesch, A.S., Sengupta, A., Som, N., Steel, A., Theobald, D.M., Torgersen, C.E., Wenger, S.J. 2013. Stream networks in space: concepts, models, and synthesis. *Ecology Letters*. doi: 10.1111/ele.12084.



Block Prediction on a Stream Network



Review of BLUP and Point Prediction

$$\begin{pmatrix} \mathbf{z} \\ Z(\mathbf{s}_0) \end{pmatrix} = \begin{pmatrix} \mathbf{X} \\ \mathbf{x}(\mathbf{s}_0)' \end{pmatrix} \boldsymbol{\beta} + \begin{pmatrix} \boldsymbol{\epsilon} \\ \epsilon(\mathbf{s}_0) \end{pmatrix}$$

$$\text{cov} \begin{pmatrix} \boldsymbol{\epsilon} \\ \epsilon(\mathbf{s}_0) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{c} \\ \mathbf{c}' & \sigma_0^2 \end{pmatrix}$$

Best Linear Unbiased Prediction (BLUP) (or [Universal] Kriging)
minimize: $E[\boldsymbol{\lambda}'\mathbf{z} - Z(\mathbf{s}_0)]^2$ subject to $E[\boldsymbol{\lambda}'\mathbf{z}] = E[Z(\mathbf{s}_0)] \forall \boldsymbol{\beta}$.

Unbiasedness $\Rightarrow \mathbf{X}'\boldsymbol{\lambda} = \mathbf{x}(\mathbf{s}_0)$

$$E[\boldsymbol{\lambda}'\boldsymbol{\epsilon}\boldsymbol{\epsilon}'\boldsymbol{\lambda} - 2\boldsymbol{\lambda}'\boldsymbol{\epsilon}\epsilon(\mathbf{s}_0) + \epsilon(\mathbf{s}_0)^2] = \boldsymbol{\lambda}'\boldsymbol{\Sigma}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}'\mathbf{c} + \sigma_0^2$$

$$\frac{\partial}{\partial \boldsymbol{\lambda}'} [\boldsymbol{\lambda}'\boldsymbol{\Sigma}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}'\mathbf{c} + \sigma_0^2 + 2\mathbf{m}'(\mathbf{X}'\boldsymbol{\lambda} - \mathbf{x}(\mathbf{s}_0))] = 0$$

$$\frac{\partial}{\partial \mathbf{m}'} [\boldsymbol{\lambda}'\boldsymbol{\Sigma}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}'\mathbf{c} + \sigma_0^2 + 2\mathbf{m}'(\mathbf{X}'\boldsymbol{\lambda} - \mathbf{x}(\mathbf{s}_0))] = 0$$

$$\Rightarrow 2\boldsymbol{\Sigma}\boldsymbol{\lambda} - 2\mathbf{c} + 2\mathbf{X}\mathbf{m} = 0, \quad \mathbf{X}'\boldsymbol{\lambda} - \mathbf{x}(\mathbf{s}_0) = 0$$

Review of BLUP and Point Prediction

solve

$$\begin{pmatrix} \Sigma & \mathbf{X} \\ \mathbf{X}' & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \mathbf{m} \end{pmatrix} = \begin{pmatrix} \mathbf{c} \\ \mathbf{x}(\mathbf{s}_0) \end{pmatrix}$$

$$\Sigma\boldsymbol{\lambda} - \mathbf{c} + \mathbf{X}\mathbf{m} = \mathbf{0} \Rightarrow \boldsymbol{\lambda} = \Sigma^{-1}(\mathbf{c} - \mathbf{X}\mathbf{m})$$

$$\mathbf{X}'\boldsymbol{\lambda} - \mathbf{x}(\mathbf{s}_0) = 0 \Rightarrow \mathbf{X}'\Sigma^{-1}(\mathbf{c} - \mathbf{X}\mathbf{m}) = \mathbf{x}(\mathbf{s}_0)$$

$$\Rightarrow \mathbf{m} = (\mathbf{X}'\Sigma^{-1}\mathbf{X})^{-1}(\mathbf{X}'\Sigma^{-1}\mathbf{c} - \mathbf{x}(\mathbf{s}_0))$$

$$\Rightarrow \boldsymbol{\lambda} = \Sigma^{-1}(\mathbf{c} + \mathbf{X}(\mathbf{X}'\Sigma^{-1}\mathbf{X})^{-1}(\mathbf{x}(\mathbf{s}_0) - \mathbf{X}'\Sigma^{-1}\mathbf{c}))$$

$$\hat{Z}(\mathbf{s}_0) = \boldsymbol{\lambda}'\mathbf{z}, \quad \text{var}(\hat{Z}(\mathbf{s}_0)) = \boldsymbol{\lambda}'\Sigma\boldsymbol{\lambda} - 2\boldsymbol{\lambda}'\mathbf{c} + \sigma_0^2$$

Alternative Formulas

$$\hat{Z}(\mathbf{s}_0) = \boldsymbol{\lambda}'\mathbf{z} = \mathbf{c}'\boldsymbol{\Sigma}^{-1}(\mathbf{z} - \hat{\boldsymbol{\mu}}) + \hat{\boldsymbol{\mu}}_0$$

where

$$\hat{\boldsymbol{\mu}} = \mathbf{X}\hat{\boldsymbol{\beta}}_{GLS} \quad \hat{\boldsymbol{\mu}}_0 = \mathbf{x}(\mathbf{s}_0)'\hat{\boldsymbol{\beta}}_{GLS} \quad \hat{\boldsymbol{\beta}}_{GLS} = (\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{z}$$

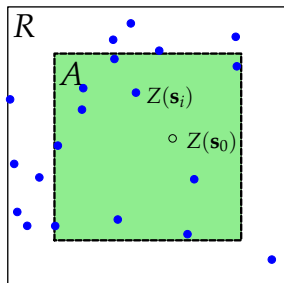
and

$$E(\boldsymbol{\lambda}'\mathbf{z} - Z(\mathbf{s}_0))^2 = \sigma_0^2 - \mathbf{c}'\boldsymbol{\Sigma}^{-1}\mathbf{c} + \mathbf{d}'(\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{d}$$

where

$$\mathbf{d} = \mathbf{x}(\mathbf{s}_0) - \mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{c}$$

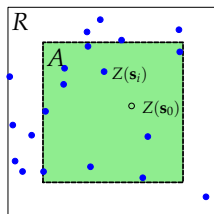
Block Prediction



- ▶ $Z(A) \equiv \int_A Z(\mathbf{u}) d\mathbf{u} / |A|$
- ▶ $x_i(A) \equiv \int_A x_i(\mathbf{u}) d\mathbf{u} / |A|$
- ▶ $\mathbf{x}_A \equiv [x_1(A), \dots, x_p(A)]'$
- ▶ $\epsilon(A) \equiv \int_A \epsilon(\mathbf{u}) d\mathbf{u} / |A|$

$$\begin{pmatrix} \mathbf{z} \\ Z(A) \end{pmatrix} = \begin{pmatrix} \mathbf{X} \\ \mathbf{x}'_A \end{pmatrix} \beta + \begin{pmatrix} \epsilon \\ \epsilon(A) \end{pmatrix}$$

Block Prediction



- ▶ $c_i(A) \equiv \text{cov}(\epsilon(\mathbf{s}_i), \epsilon(A)) = \int_A \text{cov}(\epsilon(\mathbf{s}_i), \epsilon(\mathbf{u})) d\mathbf{u} / |A|$
- ▶ $\mathbf{c}_A \equiv [c_1(A), \dots, c_n(A)]$
- ▶ $\sigma_A^2 \equiv \int_A \int_A \text{cov}(\epsilon(\mathbf{u}), \epsilon(\mathbf{v})) d\mathbf{u} d\mathbf{v} / |A|^2$

recall: $\text{cov}(Y_1, Y_2 + Y_3) = \text{cov}(Y_1, Y_2) + \text{cov}(Y_1, Y_3)$

$$\text{cov} \begin{pmatrix} \boldsymbol{\epsilon} \\ \epsilon(A) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{c}_A \\ \mathbf{c}_A' & \sigma_A^2 \end{pmatrix}$$

Block Prediction

$$\begin{pmatrix} \mathbf{z} \\ Z(A) \end{pmatrix} = \begin{pmatrix} \mathbf{X} \\ \mathbf{x}'_A \end{pmatrix} \boldsymbol{\beta} + \begin{pmatrix} \boldsymbol{\epsilon} \\ \epsilon(A) \end{pmatrix}$$

$$\text{cov} \begin{pmatrix} \boldsymbol{\epsilon} \\ \epsilon(A) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{c}_A \\ \mathbf{c}'_A & \sigma_A^2 \end{pmatrix}$$

Best Linear Unbiased Prediction (BLUP)

minimize: $E(\boldsymbol{\lambda}'\mathbf{z} - Z(A))^2$ subject to $E[\boldsymbol{\lambda}'\mathbf{z}] = E[Z(A)] \quad \forall \boldsymbol{\beta}$

Unbiasedness $\Rightarrow \mathbf{X}'\boldsymbol{\lambda} = \mathbf{x}_A$

$$E[\boldsymbol{\lambda}'\boldsymbol{\epsilon}\boldsymbol{\epsilon}'\boldsymbol{\lambda} - 2\boldsymbol{\lambda}'\boldsymbol{\epsilon}\epsilon(A) + \epsilon(A)^2] = \boldsymbol{\lambda}'\boldsymbol{\Sigma}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}'\mathbf{c}_A + \sigma_A^2$$

$$\boldsymbol{\lambda} = \boldsymbol{\Sigma}^{-1}(\mathbf{c}_A + \mathbf{X}(\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}(\mathbf{x}_A - \mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{c}_A))$$

$$\hat{Z}(A) = \boldsymbol{\lambda}'\mathbf{z}, \quad \text{var}(\hat{Z}(A)) = \boldsymbol{\lambda}'\boldsymbol{\Sigma}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}'\mathbf{c}_A + \sigma_A^2$$

Alternative Formulas

$$\hat{Z}(A) = \boldsymbol{\lambda}'\mathbf{z} = \mathbf{c}'_A \boldsymbol{\Sigma}^{-1}(\mathbf{z} - \hat{\boldsymbol{\mu}}) + \hat{\mu}_A$$

where

$$\hat{\boldsymbol{\mu}} = \mathbf{X}\hat{\boldsymbol{\beta}}_{GLS} \quad \hat{\mu}_A = \mathbf{x}'_A \hat{\boldsymbol{\beta}}_{GLS} \quad \hat{\boldsymbol{\beta}}_{GLS} = (\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{z}$$

and

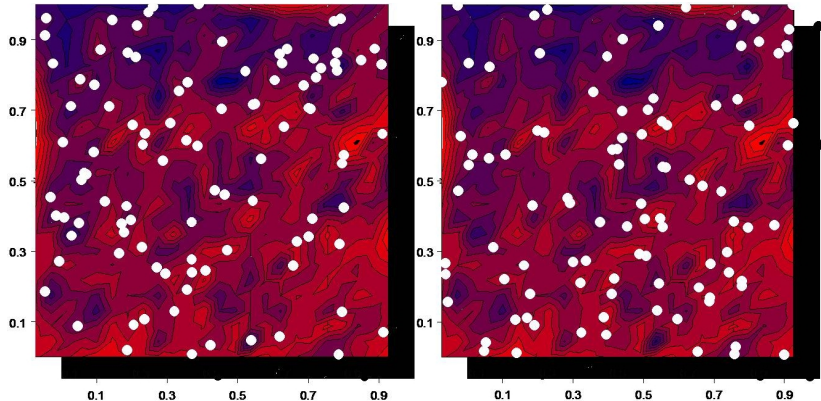
$$E(\boldsymbol{\lambda}'\mathbf{z} - Z(A))^2 = \sigma_A^2 - \mathbf{c}'_A \boldsymbol{\Sigma}^{-1}\mathbf{c}_A + \mathbf{d}'_A (\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{d}_A$$

where

$$\mathbf{d}_A = \mathbf{x}_A - \mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{c}_A$$

Fixed Pattern, Random Samples

Ver Hoef, J.M. 2002. Sampling and geostatistics for spatial data. *Ecoscience* 9: 152 - 161.

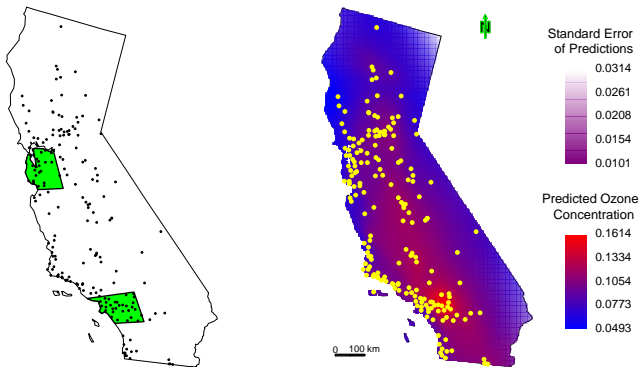


Fixed Pattern, Random Samples

1000 random samples of size 100

Validation Statistics	Simple Random Sample	Block Prediction
Bias	0.002	-0.020
RMSPE	1.28	1.02
RAEV	1.29	1.00
80%CI	0.813	0.806

Ozone Example



Ozone Example

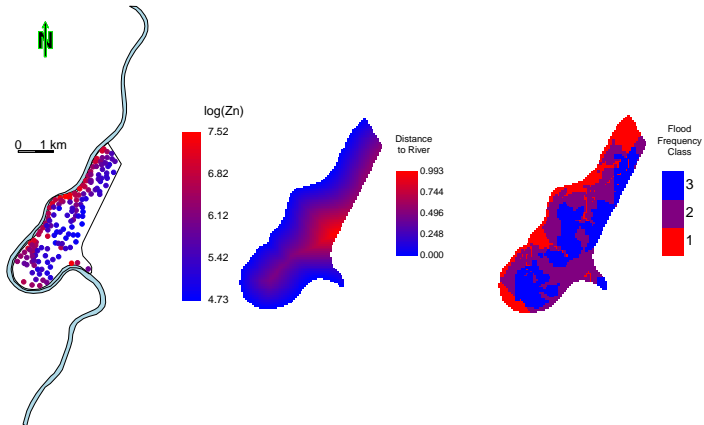
```
library(spPlotSampCourse)
library(maptools)
path <- system.file("rawdata/airPolluteCA", package = "spPlotSampCourse")
outlineFile <- paste(path, "/", "ca_outline", sep = "")
otl <- readShapePoly(outlineFile)
pointsFile <- paste(path, "/", "ca_ozone_pts", sep = "")
pts <- readShapePoints(pointsFile)
polyLAFile <- paste(path, "/", "polyLA", sep = "")
polyLA <- readShapePoly(polyLAFile)
polySFFile <- paste(path, "/", "polySF", sep = "")
polySF <- readShapePoly(polySFFile)
ozFit1 <- splmm(OZONE ~ 1, spdata = pts, estMeth = "REML", varComps = "circular",
  useAnisotropy = TRUE)
blockPredGridLA <- createBlockPredGrid(polyLA)
predictBlock(ozFit1, blockPredGridLA)

##      OZONE BlockPredSE
## 1 0.1256      0.00178

blockPredGridSF <- createBlockPredGrid(polySF)
predictBlock(ozFit1, blockPredGridSF)

##      OZONE BlockPredSE
## 1 0.08856      0.002618
```

Meuse Example



Meuse Example

```
data(meuse)
coordinates(meuse) <- ~x + y
proj4string(meuse) <- CRS("+init=epsg:28992")
data(meuse.grid)
coordinates(meuse.grid) <- ~x + y
proj4string(meuse.grid) <- CRS("+init=epsg:28992")
znFit1 <- splmm(zinc ~ dist + ffreq, spdata = meuse, varComps = "besselK")
predictBlock(znFit1, meuse.grid)

##      zinc BlockPredSE
## 1 351.9      18.73
```

Meuse Example

```
summary(znFit1)$coefficients
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    959.9      93.87   10.225 5.620e-19
## dist          -1189.6    234.71   -5.068 1.159e-06
## ffreq2         -288.7     37.26   -7.748 1.262e-12
## ffreq3         -276.1     57.92   -4.766 4.371e-06
```

```
summary(znFit1)$covparms
```

```
##   Variance Component Parameter Type   Estimate
## 1          nugget          nugget 7.858e+03
## 2        besse1K          parsil 8.486e+04
## 3        besse1K           range 8.250e+02
## 4        besse1K          extrap 7.807e-01
```

```
summary(znFit1)$R2g
```

```
## [1] 0.3965
```