Multivariate spatial models
and the multiKrig class

Stephan R. Sain, IMAGe, NCAR

ENAR Spring Meetings
March 15, 2009

Outline

• Overview of multivariate spatial regression models.
• Case study: pedotransfer functions and soil water profiles.
• The multiKrig class
  – Case study: NC temperature and precipitation.

A Spatial Regression Model

• A spatial regression model:
\[ Y = X\beta + h + \epsilon \]
\((n \times 1) = (n \times q)(q \times 1)(n \times 1)(n \times 1)\)

where
- \( E[h] = 0, \text{Var}[h] = \Sigma_h \)
- \( E[\epsilon] = 0, \text{Var}[\epsilon] = \sigma^2 I \)
- \( h \) and \( \epsilon \) are independent.

\( Y \sim N(X\beta, V), \quad V = \Sigma_h + \sigma^2 I \)

\( \hat{\beta} = (X'V^{-1}X)^{-1}X'V^{-1}Y, \quad \hat{h} = \Sigma_h V^{-1}(Y - X\hat{\beta}) \)
Multivariate Regression

- A multivariate, multiple regression model:
  \[ Y = X\beta + \epsilon \]

where
- Each of the \( n \) rows of \( Y \) represents a \( p \)-vector observation.
- Each of the \( p \) columns of \( \beta \) represent regression coefficients for each variable.
- The rows of \( \epsilon \) represent a collection of iid error vectors with zero mean and common covariance matrix, \( \Sigma \).

Multivariate Regression

- MLEs are straightforward to obtain:
  \[ \hat{\beta} = (X'X)^{-1}X'Y \]
  \[ \hat{\Sigma} = \frac{1}{n}Y'PY \]

where \( P = I - X(X'X)^{-1}X' \).

- Note that the columns of \( \hat{\beta} \) can be obtained through \( p \) univariate regressions.

Vec and Kronecker

- The Kronecker product of an \( m \times n \) matrix \( A \) and an \( r \times q \) matrix \( B \) is an \( mr \times nq \) matrix:

  \[ A \otimes B = \begin{bmatrix}
    a_{11}B & a_{12}B & \cdots & a_{1q}B \\
    a_{21}B & a_{22}B & \cdots & a_{2q}B \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{m1}B & a_{m2}B & \cdots & a_{mq}B
  \end{bmatrix} \]

- Some properties:
  \[ A \otimes (B + C) = A \otimes B + A \otimes C \]
  \[ A \otimes (B \circ C) = (A \otimes B) \circ C \]
  \[ (A \otimes B)(C \otimes D) = AC \otimes BD \]
  \[ (A \otimes B)' = A' \otimes B' \]
  \[ (A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \]
  \[ |A \otimes B| = |A|^m |B|^n \]
Vec and Kronecker

- The vec-operator stacks the columns of a matrix:
  \[ A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{vec}(A) = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \end{bmatrix} \]

- Some properties:
  \[
  \begin{align*}
  \text{vec}(AXB) &= (B' \otimes A) \text{vec} X \\
  \text{tr}(A'B) &= \text{vec}(A)' \text{vec}(B) \\
  \text{vec}(A + B) &= \text{vec}(A) + \text{vec}(B) \\
  \text{vec}(\alpha A) &= \alpha \text{vec}(A)
  \end{align*}
  \]

Multivariate Regression Revisited

- Rewrite the multivariate, multiple regression model:
  \[
  \begin{align*}
  \text{vec}(Y) &= (I_p \otimes X) \text{vec}(\beta) + \text{vec}(\epsilon) \\
  &\quad (np \times 1) \quad (np \times qp)(qp \times 1) \quad (np \times 1).
  \end{align*}
  \]

- What is \( \text{Var}[\text{vec}\epsilon] \)?
- What is the GLS estimator for \( \text{vec}(\beta) \)?

A Multivariate Spatial Model

- Extend the multivariate, multiple regression model:
  \[
  \begin{align*}
  \text{vec}(Y) &= (I_p \otimes X) \text{vec}(\beta) + \text{vec}(h) + \text{vec}(\epsilon) \\
  &\quad (np \times 1) \quad (np \times qp)(qp \times 1) \quad (np \times 1),
  \end{align*}
  \]

  where
  \[
  \begin{align*}
  \text{Var}[\text{vec}(h)] &= \Sigma_h = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1p} \\ \Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{p1} & \Sigma_{p2} & \cdots & \Sigma_{pp} \end{bmatrix} \\
  \text{Var}[\text{vec}(\epsilon)] &= \Sigma \otimes I_n
  \end{align*}
  \]
A Multivariate Spatial Model

- One simplification to the spatial covariance matrix is to use a Kronecker form:

\[ \Sigma_h = \rho \otimes K \]

where

- \( \rho \) is a \( p \times p \) matrix of scale parameters
- \( K \) is an \( n \times n \) spatial covariance.

A Multivariate Spatial Model

- Extend the multivariate, multiple regression model:

\[
\begin{align*}
vec(Y) &= (I_p \otimes X) vec(\beta) + vec(h) + vec(\epsilon) \\
&= (np \times 1) (np \times qp)(qp \times 1) (np \times 1) (np \times 1)
\end{align*}
\]

OR

\[
Y = X\beta + h + \epsilon
\]

- Now everything follows...

Case Study: Pedotransfer Functions

- Soil characteristics such as composition (clay, silt, sand) are commonly measured and easily obtainable.

- Unfortunately, crop models require water holding characteristics such as the wilting point or lower limit (LL) and the drained upper limit (DUL) which are not so easy to obtain.
  - Often the LL and DUL are a function of depth - soil water profile.
Case Study: Pedotransfer Functions

- *Pedotransfer functions* are commonly used to estimate LL and DUL.
  - Differential equations, regression, nearest neighbors, neural networks, etc.
  - Often specialized by soil type and/or region.

- Develop a new type of pedotransfer function that can capture the entire soil water profile (LL & DUL as a function of depth).
  - Characterize the variation!

Soil Water Profiles

The Big Picture

- Soil
  - Water holding characteristics
  - Bulk density
  - Etc.
- Weather (20 years)
  - Solar radiation
  - Temperature max/min
  - Precipitation

*The CERES Crop Model*
The Big Picture

- Given a complicated array of inputs, the CERES crop model will give the yields of, for example, maize.
- Deterministic output – variation in yields also of interest.
- Goals:
  - Establish a framework to study sources of variation in crop yields.
  - Assess impacts of climate change on crop yields.

Data

- \( n = 272 \) measurements on \( N = 63 \) soil samples
  - Gijsman et al. (2002)
  - Ratliff et al. (1983), Ritchie et al. (1987)
- Includes measurements of:
  - depth,
  - soil composition and texture
    * percentages of clay, sand, and silt
  - bulk density, organic matter, and
  - field measured values of LL and DUL.

- The soil texture measurements form a composition
  \[ Z_{\text{clay}} + Z_{\text{silt}} + Z_{\text{sand}} = 1 \]
  and \( Z_{\text{clay}}, Z_{\text{silt}}, Z_{\text{sand}} \) are the proportions of each soil component.
  - Not really three variables...
- To remove the dependence, the additive log-ratio transformation (Aitchinson, 1986) is applied, defining two new variables
  \[ X_1 = \log \left( \frac{Z_{\text{sand}}}{Z_{\text{clay}}} \right) \quad X_2 = \log \left( \frac{Z_{\text{silt}}}{Z_{\text{clay}}} \right). \]
A Multi-objective Pedotransfer Function

- The model for the multi-objective pedotransfer function for a particular soil is

\[ Y_0 = T_0 \beta + h(X_0) + \epsilon(D_0) \]

where

\[ Y_0 = \log \begin{bmatrix} LL_1 \\ \vdots \\ LL_d \\ \Delta_1 \\ \vdots \\ \Delta_d \end{bmatrix}, \]

and \( d \) is the number of measurements (depths) and \( \Delta_i = DUL_i - LL_i \).
A Multi-objective Pedotransfer Function

- The model for the multi-objective pedotransfer function for a particular soil is
  \[ Y_0 = T_0 \beta + h(X_0) + \epsilon(D_0) \]
  where
  \[ T_0 = \begin{bmatrix} 1 & X_0 & Z_{LL,0} & 0 \\ 0 & 1 & X_0 & Z_{\Delta,0} \end{bmatrix}, \]
  and
- \( X_0 \) is the transformed soil composition information
- \( Z_{LL} \) and \( Z_{\Delta} \) are additional covariates for \( LL \) and \( \Delta \).
  - \( Z_{LL} \) includes organic carbon
  - \( Z_{\Delta} \) includes linear and quadratic terms for depth

A Multi-objective Pedotransfer Function

- Letting
  \[ Y = \log [LL_{11} \cdots LL_{1d_1} LL_{21} \cdots LL_{Nd_2} \Delta_{11} \cdots \Delta_{1d_1} \Delta_{21} \cdots \Delta_{Nd_2}]' \]
  then \( Y \) is multivariate normal with
  \[ E[Y] = T\beta \quad \text{Var}[Y] = \Sigma_h + \Sigma_e \]
  \[ \Sigma_h = \begin{bmatrix} \rho_1 & 0 \\ 0 & \rho_2 \end{bmatrix} \otimes K \]
  \[ \Sigma_e = S \otimes R. \]
  with
  - \( K_{ij} = k(X_i, X_j) \)
  - \( S \) is the covariance of \((LL, \Delta)\) at a fixed depth
  - \( R \) is the (spatial) covariance across depths
Covariance Structures

• The covariance function for $h$ is the Matern family

$$C(d) = \sigma^2 \frac{2^{\nu d/2} K_{\nu}(\theta d)}{\Gamma(\nu)}$$

where $\sigma^2$ is a scale parameter, $\theta$ represents the range, $\nu$ controls the smoothness.

– $\sigma^2 = 1$ (the $\rho$ controls the variances), $\nu = 1$, and $\theta$ is taken to be approximately the range of the data.

– These choices represent a covariance structure that is consistent with the thin-plate spline estimator (large range, more smoothness).

– Analogous to fixing the kernel and estimating the bandwidth with kernel estimators.

Covariance Structures

• The covariance function across depths is exponential

$$C(d) = \sigma^2 \exp(-d/\theta)$$

where again $\sigma^2$ is a scale parameter and $\theta$ represents the range.

– The parameters $\sigma^2 = 1$ (the matrix $S$ controls the variances) and $\theta$ is estimated from the data.

– The multiple realizations of the soils allow for improved ability to estimate both scale and range parameters.

Covariance Structures

$$\Sigma_h = \begin{bmatrix} \rho_1 & 0 \\ 0 & \rho_2 \end{bmatrix} \otimes K$$

$$\Sigma_\epsilon = S \otimes R$$
Spatial Smoothing

- Write
  \[
  \Sigma_h + \Sigma_e = \begin{bmatrix}
  \rho_1 & 0 \\
  0 & \rho_2
\end{bmatrix} \otimes K + \begin{bmatrix}
  s_{11} & s_{12} \\
  s_{12} & s_{22}
\end{bmatrix} \otimes R \\
  = s_{11} \begin{bmatrix}
  \eta_1 & 0 \\
  0 & \eta_2
\end{bmatrix} \otimes K + \begin{bmatrix}
  1 & v_{12} \\
  v_{12} & v_{22}
\end{bmatrix} \otimes R \\
  = s_{11} \Omega
  \]

- The amount of smoothing is due to the relative contributions of the variance components, i.e. \( \eta_1 \) and \( \eta_2 \).

- Different degrees of smoothing are allowed for LL and \( \Delta \).

- Also, this construction allows for different degrees of variation in the error terms for LL and the \( \Delta \) variables.

The Estimator

- The model suggests an estimator of the form
  \[
  \hat{Y}_0 = T_0 \hat{\beta} + K'_0 \hat{\delta},
  \]
  where
  \[
  K'_0 = \begin{bmatrix}
  \eta_1 & 0 \\
  0 & \eta_2
\end{bmatrix} \otimes K.
  \]

- To fit the model, we must estimate:
  - \( \eta_1 \), \( \eta_2 \) and \( s_{11} \)
  - \( \beta \), \( \delta \)
  - \( R \) and the other entries of \( S \)

REML

- Take the QR decomposition of \( T \)
  \[
  T = [Q_1 \ Q_2] \begin{bmatrix}
  R \\
  0
\end{bmatrix}.
  \]

- Then \( Q_2'Y \) has zero mean and covariance matrix given by
  \[
  Q_2'(\Sigma_h + \Sigma_e)Q_2.
  \]

- Maximize (numerically) the likelihood based on \( Q_2'Y \) which is only a function of the covariance parameters.

- Estimates of \( \beta \) and \( \delta \) follow directly
  \[
  \hat{\beta} = (T'\hat{\Omega}^{-1}T)^{-1}T'\hat{\Omega}^{-1}Y \quad \hat{\delta} = \hat{\Omega}^{-1}(Y - T\hat{\beta}).
  \]
An Iterative Approach

0. Initialize: compute $K$ and set $S = I$ and $R = I$.

1. Estimate $\eta_1$ and $\eta_2$ (and $s_{11}$) via a simplified type of REML (grid search).

2. Then

$$\hat{\beta} = (T'\hat{\Omega}^{-1}T)^{-1}T'\hat{\Omega}^{-1}Y \quad \hat{\delta} = \hat{\Omega}^{-1}(Y - T\hat{\beta}).$$

3. Compute residuals and
   a. Update $S$ (R fixed) – closed form solution.
   b. Update $R$ (S fixed) – grid search for $\theta$.

4. Repeat items 1-3 until convergence.

An Iterative Approach

- Let $Y = \mu + h + \epsilon$, where $h$ and $\epsilon$ are independent Gaussian random variables; the conditional distribution of $Y - \mu - h$ given $h$ is a zero mean Gaussian with covariance matrix $\epsilon$.

- Thus, the log-likelihood associated with the residuals is given by

$$-\frac{n}{2}|S| - |R| - \text{vec}(U)'(S^{-1} \otimes R^{-1})\text{vec}(U)$$

- The quadratic form can be written as

$$\text{tr}(S^{-1}\sum_i\sum_j r_{ij}u_i'u_j')$$

where $r_{ij}$ is the $ij$th element of $R^{-1}$ and $u_i$ is the bivariate, unstacked residual for the $i$th observation.

An Iterative Approach

- An update for $S$ can be written as

$$\hat{S} = \frac{1}{n}\sum_i\sum_j r_{ij}u_i'u_j'$$

$$= \frac{1}{n}\text{vec}(U)'R^{-1}\text{vec}(U)$$

where $U$ is the $n \times 2$ matrix of unstacked residuals.

- Again, a simple grid search for $\theta$ is used to obtain a new value for $R$. 
Parameter Estimates

<table>
<thead>
<tr>
<th>Method</th>
<th>$\eta_1$</th>
<th>$\eta_2$</th>
<th>$S_{11}$</th>
<th>$S_{22}$</th>
<th>$S_{12}$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>REML</td>
<td>5.84</td>
<td>1.66</td>
<td>0.0765</td>
<td>0.0483</td>
<td>-0.0222</td>
<td>134.6</td>
</tr>
<tr>
<td>Iterative</td>
<td>5.74</td>
<td>2.21</td>
<td>0.0697</td>
<td>0.0445</td>
<td>-0.0217</td>
<td>144.2</td>
</tr>
</tbody>
</table>

Soil Composition and LL

![Soil Composition and LL](image1.png)

Soil Composition and $\Delta$

![Soil Composition and $\Delta$](image2.png)
Soil Composition and LL/Δ

Organic Carbon and LL

Depth and Δ
Residuals (Within Depth)

Spatial Covariance Across Depth

Prediction Error

- The real benefit of considering our estimator as a spatial process is in the interpretation with respect to the uncertainty of the estimator:
  - The thin-plate spline is a biased estimator with uncorrelated error; not easy to quantify the bias (interpolation error and smoothing error).
  - The spatial process estimator is unbiased, but with correlated error; more complicated error structure but conceptually straightforward to work with.
Prediction Error

- The estimator can be written as

\[
\hat{Y}_0 = T_0\hat{\beta} + K_0\hat{\delta} = A_0Y,
\]

where

\[
A_0 = T_0(T'\hat{\Omega}^{-1}T)^{-1}T'\hat{\Omega}^{-1} + K_0(\hat{\Omega}^{-1} - \hat{\Omega}^{-1}T(T'\hat{\Omega}^{-1}T)^{-1}T'\hat{\Omega}^{-1}).
\]

Prediction Error

- Hence,

\[
\text{Var}(Y_0 - \hat{Y}_0) = \text{Var}(Y_0 - A_0Y) = \text{Var}(Y_0) + A_0\text{Var}(Y)A_0' - 2A_0\text{Cov}(Y, Y_0).
\]

- \(\text{Var}(Y_0)\) and \(\text{Var}(Y)\) are computed by plugging in parameters estimates for \(\Sigma_h\) and \(\Sigma_e\).

- The covariance between \(Y_0\) and \(Y\) comes from \(h\) and is based on the distance between the transformed composition data.

Generation of Soil Profiles

- Simulations of log \(LL\) and log \(\Delta\) were generated from a multivariate normal with mean \(A_0Y\) and variance given by the prediction error.

- We use an average soil composition profile computed from the data and assumed to constant across all depths,

\[D = \{5, 15, 30, 45, 60, 90, 120, 150\}.
\]

- Represent a draw from the posterior distribution of soil water profiles based on the estimated mean and covariance structure and the uncertainty gleaned from the data.
Generation of Soil Profiles

Application: Crop Models

- Two soils (SIL, S)
  - Given the soil composition, organic carbon, depth, etc., 100 soil profiles (LL, DUL) were generated.

- Twenty years of weather (solar radiation, temperature min/max, and precipitation).

- Yield output generated from the CERES-Maize crop model.

Crop Yields

- SIL (red), S (blue), total annual precipitation (solid line)
Crop Yields

- SIL (red), S (blue), average annual temperature (solid line)

The multiKrig Class

- Create a multivariate spatial regression within the univariate Krig framework:

\[ Y = T\beta + h + \epsilon \]

\[
Y = \begin{bmatrix}
Y_1 \\
\vdots \\
Y_p
\end{bmatrix}
\quad T = I_p \otimes X 
\quad h = \begin{bmatrix}
h_1 \\
\vdots \\
h_p
\end{bmatrix}
\quad \epsilon = \begin{bmatrix}
\epsilon_1 \\
\vdots \\
\epsilon_p
\end{bmatrix}
\]

The multiKrig Class

- Create a multivariate spatial regression within the univariate Krig framework:

\[
E[Y] = T\beta
\]
\[
\text{Var}[Y] = \Sigma_h + \Sigma_\epsilon
\]
\[
= \begin{bmatrix}
\rho_1 & \cdots & \rho_p \\
& \ddots & \\
& & \rho_p
\end{bmatrix} \otimes V(\theta) + 
\begin{bmatrix}
s_{11} & s_{12} & \cdots & s_{1p} \\
s_{21} & s_{22} & \cdots & s_{2p} \\
\vdots & \vdots & \ddots & \vdots \\
s_{p1} & s_{p2} & \cdots & s_{pp}
\end{bmatrix} \otimes I_n
\]
\[
= \rho_1 R \otimes V(\theta) + s_{11} S \otimes I_n
\]
\[
= \rho_1 (R \otimes V(\theta) + \lambda S \otimes I_n)
\]
The multiKrig Class

- Create a multivariate spatial regression within the univariate Krig framework:
  \[ Y = T\beta + h + \epsilon \]
  \[ E[Y] = T\beta \]
  \[ \text{Var}[Y] = \Sigma h + \Sigma \epsilon \]
  \[ = \rho_1 (R \otimes V(\theta) + \lambda S \otimes I_n) \]

- Given \( S, R, \) and \( \theta \), use Krig to estimate \( \beta, \rho_1 \) and \( \lambda \).

The multiKrig Class

- Issues:
  - Specifying \( x, Y, \) and \( Z \)
  - Mean function (null.function)
  - Covariance function (cov.function)
  - Error function (wght.function)

- Estimation (\( S, R, \) and \( \theta \))

Krig Function

\[
\text{Krig} <- \text{function}(x, Y, Z, \\
\text{null.function = "Krig.null.function",} \\
\text{cov.function = "stationary.cov",} \\
\text{wght.function = NULL,} \\
\text{null.args = NULL, cov.args = NULL, wght.args = NULL})
\]

- \( x \) is an \( n \times q \) matrix of spatial locations
- \( Y \) is a \( n \)-vector of observations
- \( Z \) is a \( n \times q \) matrix of additional covariates
multiKrig Function

multiKrig <- function(s,Y,Z,
    cov.function="multi.cov",cov.args=NULL,
    wght.function="multi.wght",wght.args=NULL)

• \(s\) is an \(n \times q\) matrix of spatial locations
• \(Y\) is an \(n \times p\) matrix of observations
• \(Z\) is either:
  – a \(n \times q\) matrix of additional covariates, or
  – a list of \(n \times q_i\) matrices of additional covariates
x <- expand.grid(1:n,1:d)

\[
\begin{bmatrix}
1 & 1 \\
2 & 1 \\
n & 1 \\
n & 2 \\
\end{bmatrix}
\]

multiKrig <- function(s,Y,Z,
cov.function="multi.cov",cov.args=NULL,
wght.function="multi.wght",wght.args=NULL)
{
  d <- ncol(Y)
  n <- nrow(Y)
  Y <- c(Y)
  x <- expand.grid(1:n,1:d)
  nZ <- kronecker(diag(d),cbind(s,Z))
  obj <- Krig(x=x,Y=c(Y),Z=nZ,method="REML",
null.function="multi.null",
cov.function=cov.function,cov.args=cov.args,
wght.function=wght.function,wght.args=wght.args)
}

nZ <- kronecker(diag(d),cbind(s,Z))

\[
\begin{bmatrix}
s & Z \\
\vdots & \ddots & s & Z \\
\end{bmatrix}
\]

multi.null <- function(x,Z=NULL,drop.Z=FALSE)
{
data <- data.frame(a=as.factor(x[,2]))
X <- model.matrix(~a,data=data,contrasts=list(a="contr.treatment"))
return(cbind(X,Z))
}

T = \[
\begin{bmatrix}
1 & 0 & 0 & s & Z & 0 & 0 \\
1 & 1 & 0 & s & Z & 0 \\
1 & 0 & 1 & 0 & 0 & s & Z \\
\end{bmatrix}
\]

multiKrig <- function(s,Y,Z,
cov.function="multi.cov",cov.args=NULL,
wght.function="multi.wght",wght.args=NULL)
{
  obj <- Krig(x=x,Y=c(Y),Z=nZ,method="REML",
cov.function=cov.function,cov.args=cov.args,
wght.function=wght.function,wght.args=wght.args)
}
Covariance Function

• Issue: $x$ is now a matrix of indices.

• Solution: pass the spatial locations as an argument to the covariance function.

```r
multiKrig <- function(s,Y,Z,
cov.function="multi.cov",cov.args=NULL,
wght.function="multi.wght",wght.args=NULL)
{
  cov.args$x <- s
  obj <- Krig(x=s,y=Y,method="REML",
cov.function=cov.function,cov.args=cov.args,
wght.function=wght.function,wght.args=wght.args)
}

multi.cov <- function(x1,x2,marginal=FALSE,C=NA,
theta,smoothness)
{
  ind <- unique(x1[,2])
temp <- stationary.cov(s[x1[,1][x1[,2]==1],],
s[x2[,1][x2[,2]==1],],
Covariance="Matern",theta=theta,smoothness=smoothness)
  if (length(ind)==1)
    for (i in 2:length(ind))
      temp2 <- rho[i-1]*stationary.cov(s[x1[,1][x1[,2]==i],],
s[x2[,1][x2[,2]==i],],
Covariance="Matern",theta=theta,smoothness=smoothness)
  d1 <- dim(temp)
  d2 <- dim(temp2)
  temp <- rbind(cbind(temp,matrix(0,d1[1],d2[2])),
cbind(matrix(0,d1[1],d2[2]),temp2))
}
return(temp)
}
```

Weight Function

```r
multiKrig <- function(s,Y,Z,
cov.function="multi.cov",cov.args=NULL,
wght.function="multi.wght",wght.args=NULL)
{
  obj <- Krig(x=s,y=Y,method="REML",
cov.function="multi.null",
cov.function=cov.function,cov.args=cov.args,
wght.function=wght.function,wght.args=wght.args)
}

multi.wght <- function(x,sp)
{
  n <- length(unique(x[,1]))
  return(kronecker(solve(S),diag(n)))
}
```
Estimation

• Krig will estimate $\beta$, $\rho_1$ and $\lambda$.
  - REML
  - GCV (not quite there...)

• How to estimate $S$, $R$, and $\theta$?

Thanks!

sain@ucar.edu
www.image.ucar.edu/~ssain