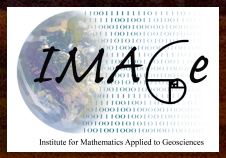
Applied Spatial Statistics: Lecture 6 Multivariate Normal

Douglas Nychka,
National Center for Atmospheric Research



Supported by the National Science Foundation Boulder, Spring 2013

Outline

- additive model
- Multivariate normal by a linear transformation
- Likelihood and log Likelihood
- closed forms for maximizing over ρ and $\boldsymbol{d}$
- conditional density functions
- application to the spatial statistics problem.

Estimating a curve or surface.

The additive statistical model:

Given n pairs of observations (x_i, y_i), $i = 1, \ldots, n$

$$\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\epsilon}$$

$\boldsymbol{\epsilon}_i$'s are random errors $E[\boldsymbol{\epsilon}] = 0$ COV($\boldsymbol{\epsilon}$) $= \sigma^2 I$

and f is an unknown smooth function:

$$f(\boldsymbol{x}) = \sum_{j=1}^{p} \phi_j(\boldsymbol{x}) \boldsymbol{d}_j + g(\boldsymbol{x})$$

- $\{\phi_j\}$ *known* basis functions based on spatial locations

e.g. polyonimials or topography. $\boldsymbol{d}$ fixed coefficients but not known..

- $g(\boldsymbol{x})$ a Gaussian process expected value of 0 and a covariance function $\rho k_\theta(\boldsymbol{x}_1, \boldsymbol{x}_2)$ form of k is *known* expect for some parameter values: ρ and θ that are unknown.

A minimum set of parameters:

- fixed effect linear parameters $\boldsymbol{d}$.
- nugget, error variance σ^2
- sill, process variance ρ
- other covariance parameters such as range, scale θ

The goal is to identify the statistical likelihood so we can estimate these (and possibly others).

Folklore:

$\boldsymbol{d}$ is easy – just regression

$\lambda = \sigma^2/\rho$ and σ^2 are more difficult with no closed formulas

ρ more sensitive to data and covariance shape.

θ This is hard to estimate with small sample sizes.

Multivariate normal construction

$\boldsymbol{u}$: a vector of N independent N(0,1)s

$\boldsymbol{u} = u_1, u_2, ..., u_N$ independent $u_k \sim$ Normal $(0, 1)$.

joint probability density function:

$$h(\boldsymbol{u}) = \frac{1}{(2\pi)^{N/2}} e^{-(1/2)\sum_{k=1}^{N} u_k^2} = \frac{1}{(2\pi)^{N/2}} e^{-\frac{\boldsymbol{u}^T\boldsymbol{u}}{2}}$$

Def 1 $\boldsymbol{v} = A\boldsymbol{u} + \boldsymbol{\mu}$ is multivariate normal with expectation μ and covariance $\Sigma = AA^T$

Def 2

probability density function for $\boldsymbol{v}$ is

$$h(\boldsymbol{v}) = \frac{1}{(2\pi)^{N/2}} e^{-\frac{(\boldsymbol{v}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{v}-\boldsymbol{\mu})}{2}} |\Sigma|^{-1/2}$$

BTW

This result is "if and only if" Def 1 $\Rightarrow$ Def 2 and Def 1 $\Rightarrow$ Def2

Why does this work?

In general if $T(\boldsymbol{v}) = \boldsymbol{u}$ is any transformation of a random vector then the probability density function for $\boldsymbol{v}$ is

$$h(T(\boldsymbol{v}))J(\boldsymbol{v})$$

J is the Jacobian term, the determinant (indicated by $|.|$) of the matrix of partial derivatives:

$$J(\boldsymbol{v}) = \begin{vmatrix} \frac{\partial T(\boldsymbol{u}_1)}{\boldsymbol{v}_1} & \frac{\partial T(\boldsymbol{u}_1)}{\boldsymbol{v}_2} & \cdots & \frac{\partial T(\boldsymbol{u}_1)}{\boldsymbol{v}_N} \\ \frac{\partial T(\boldsymbol{u}_2)}{\boldsymbol{v}_1} & \frac{\partial T(\boldsymbol{u}_2)}{\boldsymbol{v}_2} & \cdots & \frac{\partial T(\boldsymbol{u}_2)}{\boldsymbol{v}_N} \\ \vdots & & & \\ \frac{\partial T(\boldsymbol{u}_N)}{\boldsymbol{v}_1} & \frac{\partial T(\boldsymbol{u}_N)}{\boldsymbol{v}_2} & \cdots & \frac{\partial T(\boldsymbol{u}_2)}{\boldsymbol{v}_N} \end{vmatrix} \quad (1)$$

- Stats students suffer through devilishly hard " change of variables" problems for their qualifiers where T is complex and nonlinear

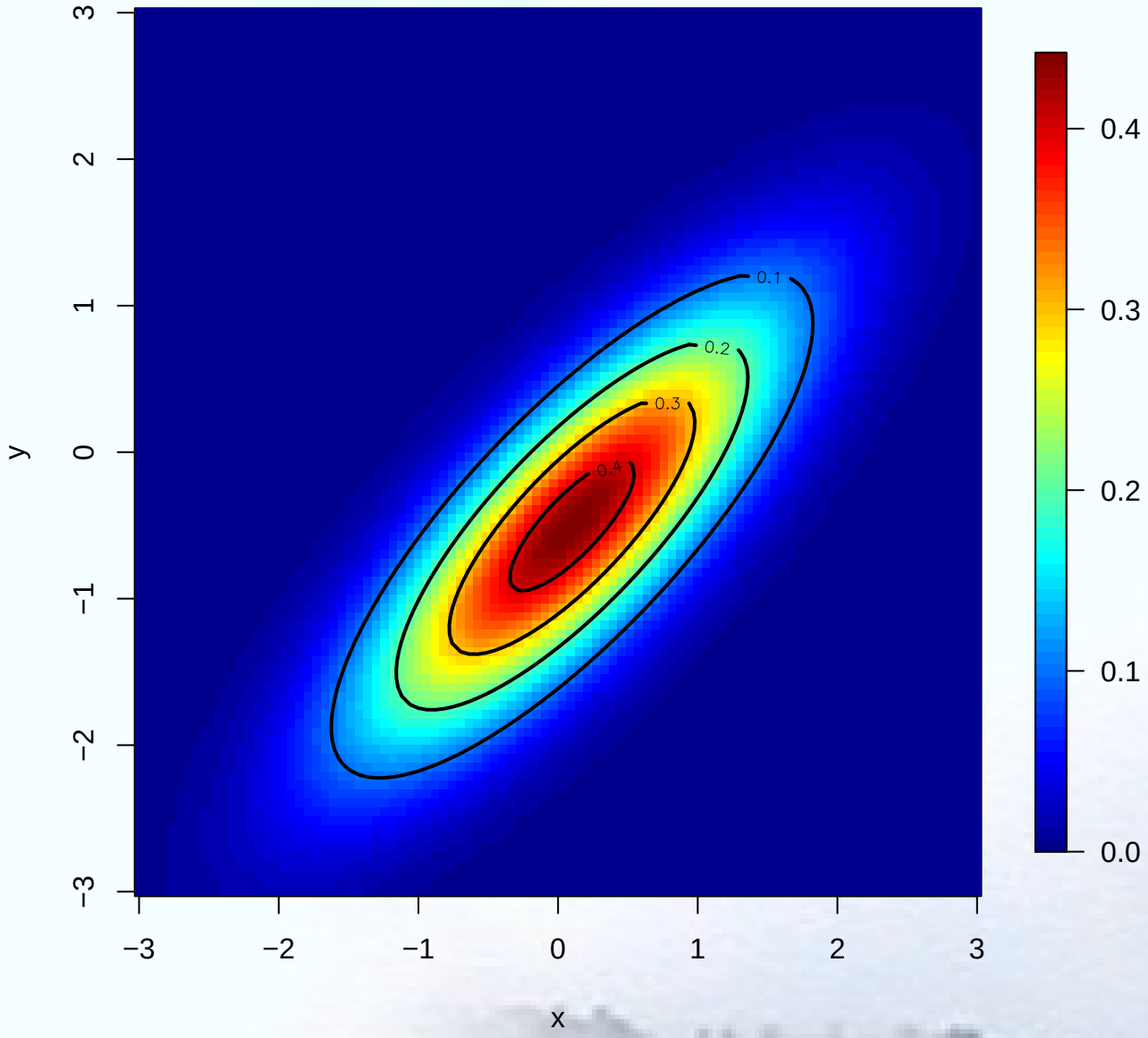
- When T is linear: $T(\boldsymbol{v}) = A^{-1}(\boldsymbol{v} - \boldsymbol{\mu}) = \boldsymbol{u}$ the Jacobian is filled with constant elements and the determinant is simplified by some identities.

Multivariate normal density in R

A 2-d example but reviewing matrix/vector techniques in R.

```
mu<- matrix( c( .1 , -.5), ncol=1, nrow=2)
Sigma<-  rbind(  c(1,.8),
                          c(.8,1) )
v.grid<- make.surface.grid(
                   list( x= seq( -3,3,,100), y =   seq( -3,3,,100) ))
z<- rep( NA, 100*100)
for( k in 1:(100*100)){
        v<-  v.grid[k,] # a vector with 2 rows
        z[k] <-  ( 1/ (2*pi)^( 2/2) ) *
              exp(  - t(v - mu) %*% solve( Sigma) %*% (v - mu)/2 ) *
                                    det( Sigma)^(-2/2)
 }
 surface( as.surface( v.grid, z))
```

the density surface



Likelihood for covariance parameters

Combine the random surface with the measurement error component

$$\boldsymbol{y} \sim MN(X\boldsymbol{d}, \rho K_\theta + \sigma^2)$$

- X the fixed part of the model
- K covariance matrix for the observations based on covariance function'' :

$$K_{i,j} = k_\theta(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

e.g. $K_{i,j} = \rho e^{-}||\boldsymbol{x}_i - \boldsymbol{x}_j||/\theta$ for the exponential

probability density function for $\boldsymbol{y}$

$$h(\boldsymbol{y}) = \frac{1}{(2\pi)^{N/2}} \exp - \frac{(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{d})^T \left(\rho K_\theta + \sigma^2 I\right)^{-1} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{d})}{2} |\rho K_\theta + \sigma^2 I|^{-1/2}$$

log Likelihood

$$logL(\boldsymbol{y}, \rho, \sigma, \theta, \boldsymbol{d}) = -(N/2)\log(2\pi) - \left[\frac{(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{d})^T \left(\rho K_\theta + \sigma^2 I\right)^{-1} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{d})}{2}\right]$$

$$-(1/2)\log|\rho K_\theta + \sigma^2 I|$$

- First maximize over $\boldsymbol{d}$ – closed form expression: $\hat{\boldsymbol{d}}$ generalized least squares estimator.

- Given $\hat{\boldsymbol{d}}$ maximize over the other parameters – keeping in mind for every new set of parameter values $\hat{\boldsymbol{d}}$ needs to be recomputed.

simplifications

- First maximize over $\boldsymbol{d}$ – closed form expression: $\hat{\boldsymbol{d}}$ generalized least squares estimator.

- Given $\hat{\boldsymbol{d}}$ maximize over the other parameters – keeping in mind for every new set of parameter values $\hat{\boldsymbol{d}}$ needs to be recomputed.

- It is also possible to maximize over the ρ parameter in closed form along with $\boldsymbol{d}$.

$$logL(\boldsymbol{y}, \rho, \lambda, \theta, \boldsymbol{d}) = -(N/2)\log(2\pi)$$

$$-(1/\rho)\left[\frac{(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{d})^T(K_\theta+\lambda I)^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{d})}{2}\right]$$

$$-(1/2)\log|(K_\theta+\lambda I)| - (N/2)\log(\rho)$$

For fixed λ and θ easy to show

$$\hat{\rho} = (1/N)\left[(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{d}})^T(K_\theta+\lambda I)^{-1}(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{d}})\right]$$

The profiled likelihood

Substituting the profile MLEs in for $\boldsymbol{d}$ and ρ leaves

$$logL(\boldsymbol{y}, \widehat{\rho}, \lambda, \theta, \widehat{\boldsymbol{d}}) =$$

$$-(N/2)\log(2\pi) - N/2 - (1/2)\log|K_\theta + \lambda I| - (N/2)\log(\widehat{\rho})$$

Keep in mind that $\widehat{\rho}$ and $\widehat{\boldsymbol{d}}$ are implicitly functions of the remaining parameters.

Even when we know the parameters we still have to do spatial prediction . . .

Big Idea

Find the conditional distribution of the spatial field given the observations.

- *The best estimate is the conditional mean,*
  uncertainty the conditional variance.
- We will get the same answer as Kriging!

Conditional distributions

$f(x, y)$ the joint pdf for $(X, Y)$and suppose that g(x) is the pdf just for X.

$$f(y|x) = \frac{f(x, y)}{g(x)}$$

"The distribution of Y *given* the value for X. "

Here $X = x$ is observed (fixed) and we have a distribution for Y.

Note that the conditional density is *proportional* to the joint but with the conditioning variable held fixed.

A useful property of Multivariate normals is that the conditional distributions are also normal.

The two variable normal

For the bivariate normal v_1, v_2 with

- expected value μ_1, μ_2,
- variances σ_1^2, σ_2^2
- correlation α.

$f(v_2|v_1)$ is Normal

mean : $\mu_2 + \alpha\sigma_2/\sigma_1(v_1 - \mu_1)$,

variance: $\sigma_2^2(1 - \alpha^2)$

(with some simplification)

Some useful notation for pdfs:

- $[Y]$ the pdf for the random variable Y
- $[X, Y]$ pdf for joint distribution of X and Y
- $[Y|X]$ conditional pdf for Y *given* X

So the formula for the conditional is:

$$[Y|X] = [X, Y]/[X]$$

Also note that $[X, Y] = [Y|X][X]$

Bayes Theorem

Bayes Theorem gives a way of inverting the conditional information. In bracket notation it is just

$$[Y|X] = \frac{[X|Y][Y]}{[X]}$$

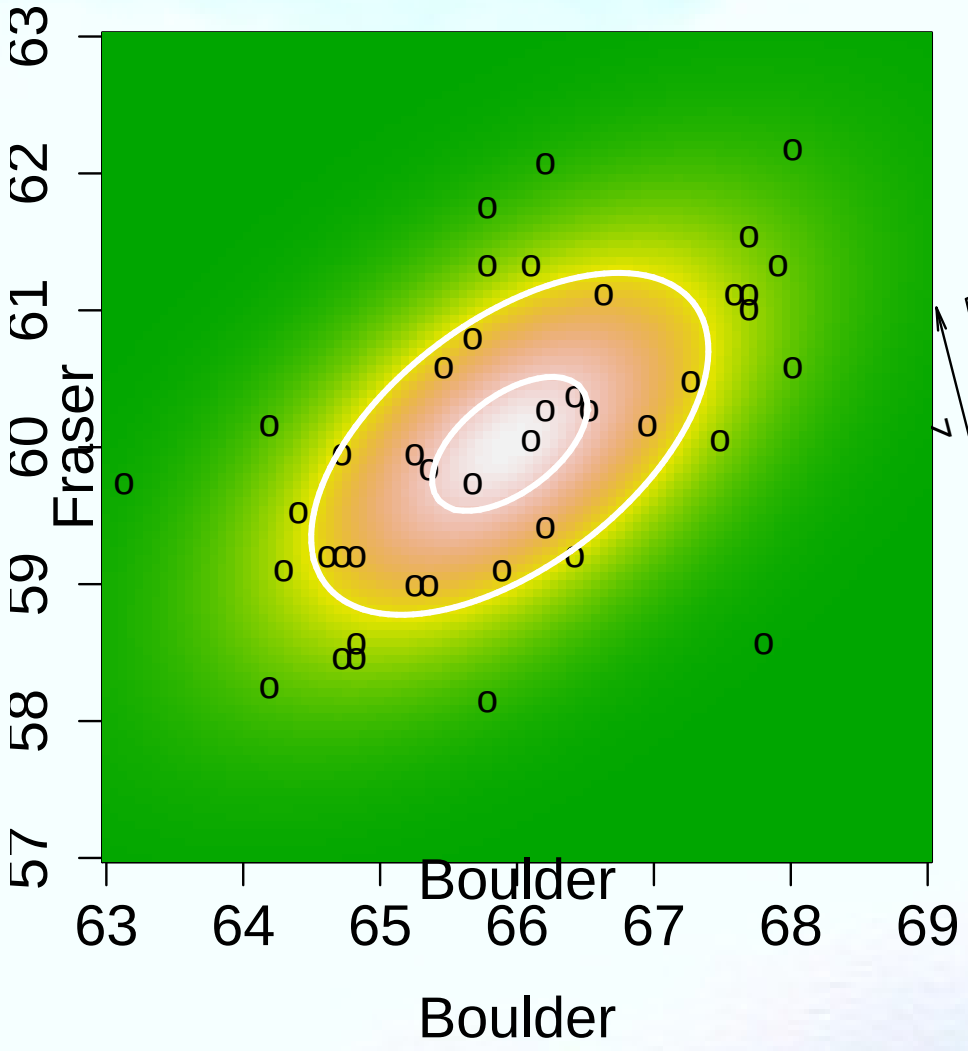
The proof follows by definitions:

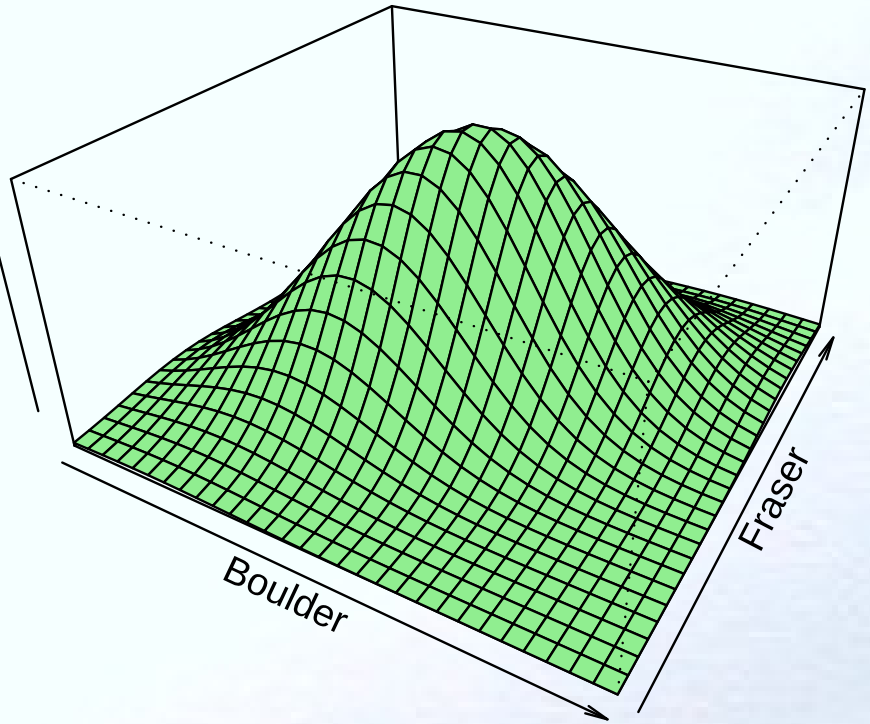
$$[Y|X] = \frac{[X,Y]}{[X]} = \frac{[X|Y][Y]}{[X]}$$

Note that $[Y|X]$ is simply proportional to the joint density where the normalization depends on the values of X. (But in many cases the normalization is difficult to find.)

Will come back to this in some later lectures

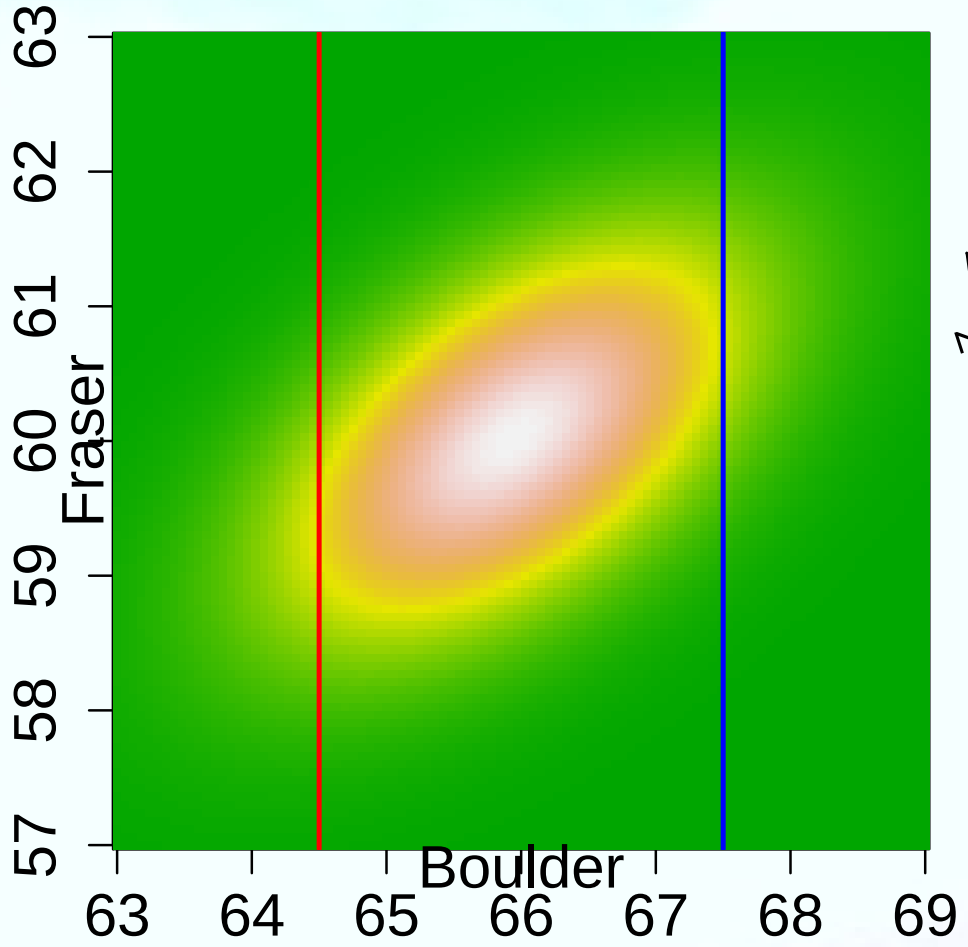
Boulder/Fraser data

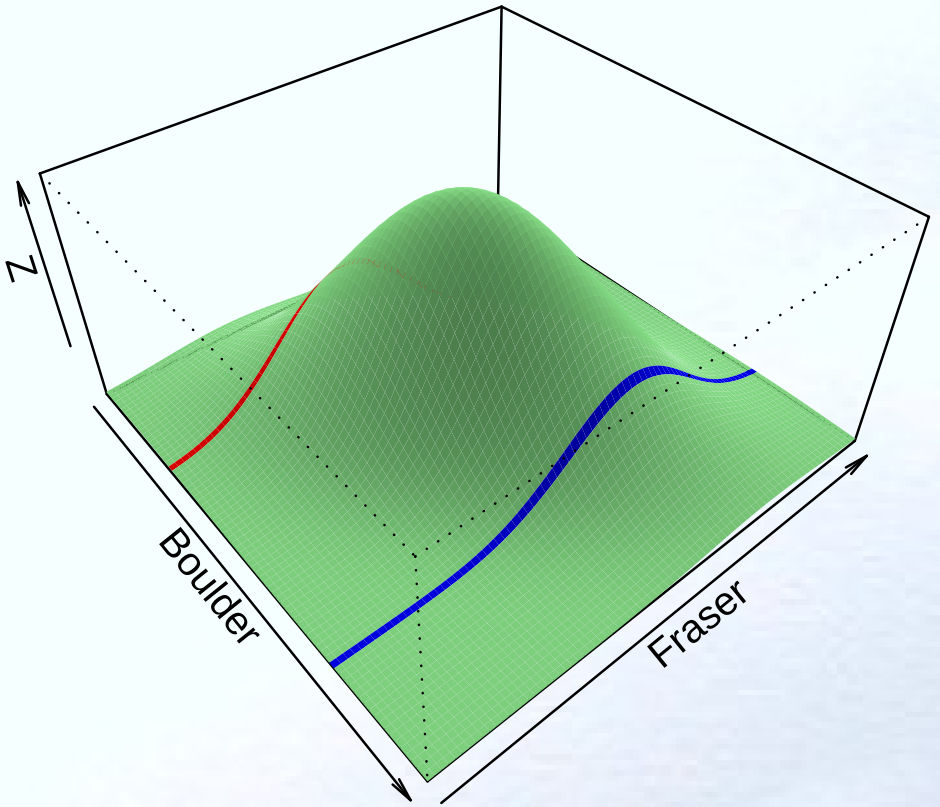




joint pdf

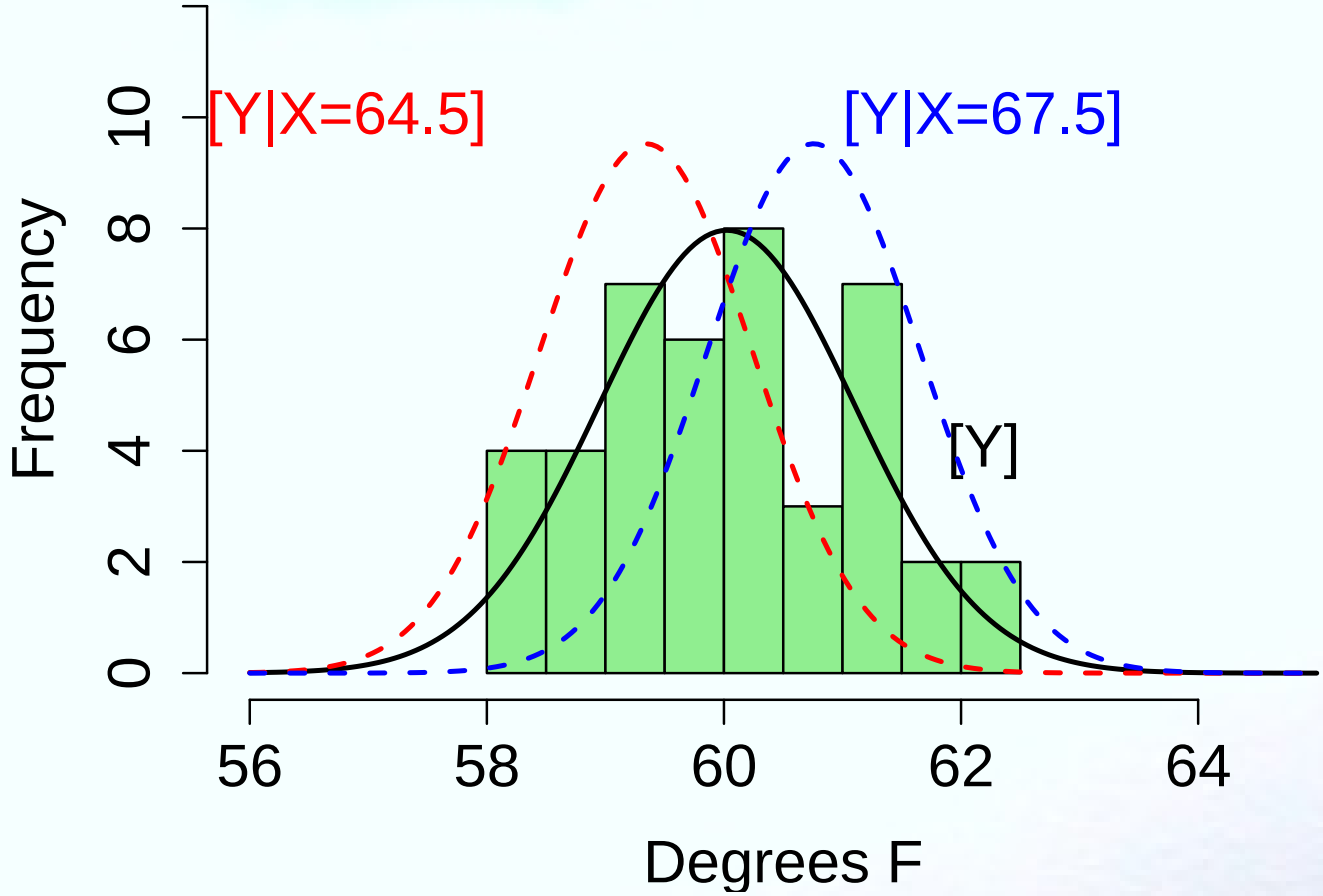
Slicing the surface





Conditional densities for Boulder/Fraser

(Y is Fraser temps and X is Boulder)



Conditional normal general case

Divide $\boldsymbol{v}$ into two parts
(e.g. what we observe and what we want to predict.)

$$\boldsymbol{v} = \begin{pmatrix} \boldsymbol{v}_1 \\ \hline \boldsymbol{v}_2 \end{pmatrix} \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \hline \boldsymbol{\mu}_2 \end{pmatrix} \quad \Sigma = \left(\begin{array}{c|c} \Sigma_{11} & \Sigma_{12} \\ \hline \Sigma_{21} & \Sigma_{22} \end{array}\right)$$

Σ is divided up so covariance for $\boldsymbol{v}_1$ is Σ_{11}, covariance for $\boldsymbol{v}_2$ is Σ_{22}

Conditional formula:

$$[\boldsymbol{v}_2|\boldsymbol{v}_1] = N(\boldsymbol{\mu}_2 + \Sigma_{2,1}\Sigma_{1,1}^{-1}(\boldsymbol{v}_1 - \boldsymbol{\mu}_1),$$
$$\Sigma_{2,2} - \Sigma_{2,1}\Sigma_{1,1}^{-1}\Sigma_{1,2})$$

- "distribution of $\boldsymbol{v}_2$ *given* $\boldsymbol{v}_1$"
- conditional mean depends on $\boldsymbol{v}_1$
- conditional covariance does not depend on $\boldsymbol{v}_1$

Our application is

$\boldsymbol{v}_1 = \boldsymbol{y}$ **(the Data),** $\boldsymbol{\mu}_1 = X\boldsymbol{d}$
and
$\boldsymbol{v}_2 = \{g(\boldsymbol{x}_1^*), ...g(\boldsymbol{x}_N^*)\}$ **(points to predict)** $\boldsymbol{\mu}_2 = 0$
spatial field values where we would like to predict. These can be either at the observation locations or at other points.

Sorting through pieces of Σ

$\Sigma_{1,1} = cov(\boldsymbol{y}, \boldsymbol{y}) = \rho K + \sigma^2 I$

i, j element of $\Sigma_{2,1} = cov(\boldsymbol{y}, \boldsymbol{g})$ is $\rho k(\boldsymbol{x}_i, \boldsymbol{x}_k^*)$

j, k element of $\Sigma_{2,2} = cov(\boldsymbol{g}, \boldsymbol{g})$ is $\rho k(\boldsymbol{x}_j^*, \boldsymbol{x}_k^*)$

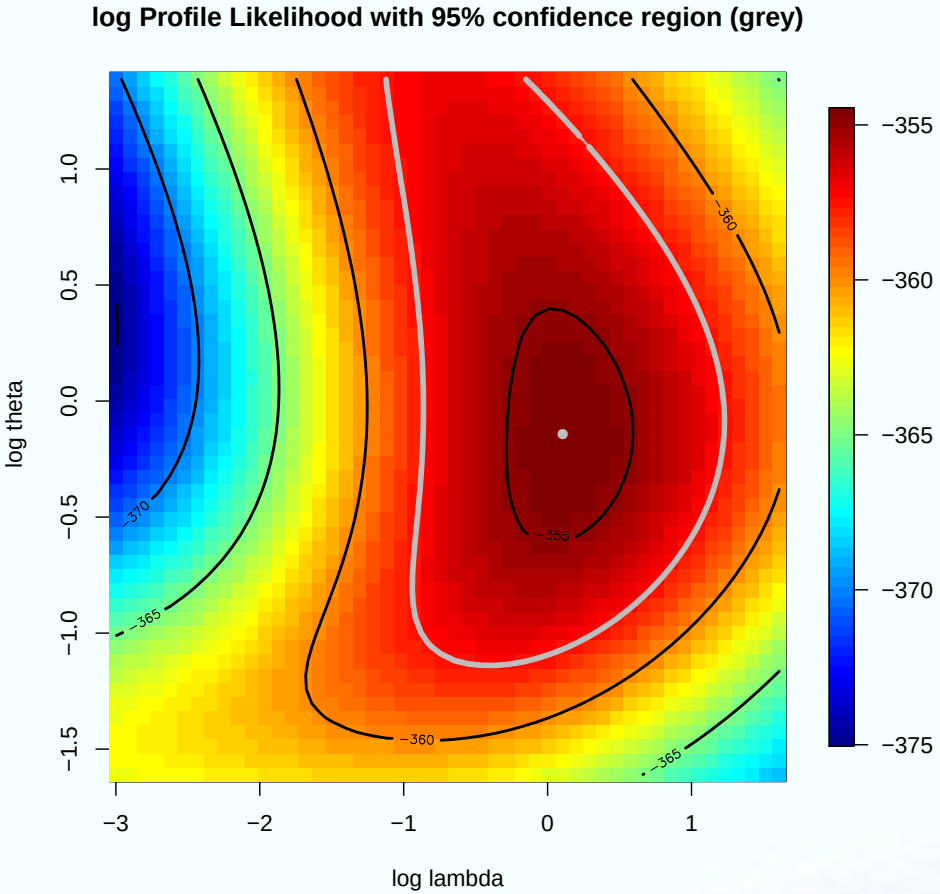
Substituting these into the conditional mean and covariance formulas gives us the Kriging estimator

Summary

- Identify a fixed part of the spatial model and the covariance.
- Estimate the parameters by maximum likelihood – possibly profiling to make computations easier
- Find conditional mean for $g(\boldsymbol{x}^*)$ using MLE for parameters
- Find prediction for fixed part of model e.g. $\sum_j \phi(\boldsymbol{x}^*)\hat{\boldsymbol{d}}_j$
- Add two together for spatial prediction at $\boldsymbol{x}^*$
- Use Kriging covariance as the uncertainty – includes uncertainty for $\hat{\boldsymbol{d}}$

Colorado minimum MAM temps.

Exponential covariance search over λ and θ



MLEs: $\theta = 0.867$, $\lambda = 1.111$ . . . and from profiling $\rho = 0.296$, $\sigma = 1.019$

Spatial estimate for Lyons, CO

(See R script to find the d and c coefficients explicitly)
Recall $\widehat{d}$ found by GLS and $\widehat{c}$ found by "Kriging" GLS residuals.

```
x.star<- rbind( c(-105.2708,40.2247) ) # Lyons, CO
data(RMelevation)
elev.star<- interp.surface(RMelevation, x.star)
X.star<- matrix(c( 1, x.star, elev.star), nrow=1)
#
fixed.part<-  X.star%*%d.MLE
ghat.x.star<- t( exp( -rdist( xHW1,x.star)/theta.MLE))%*%c.MLE
#NOTE: lambda format so left out rho in formula
Lyons.prediction <- fixed.part + ghat.x.star
```

Estimate: -0.283 degrees C

Using `fields` as a black box

```
obj<- mKrig(x = xHW1, y = yHW1, lambda = lambda.MLE,
                                        Z = elevHW1, theta = theta.MLE)
predict( obj, x= x.star, Z= elev.star)
```