Introduction to Bayes and non-Bayes spatial statistics

Gabriel Huerta

Department of Mathematics and Statistics
University of New Mexico
http://www.stat.unm.edu/∼ghuerta/geoRcode.txt

General Concepts

- *Spatial Statistics* are methods and techniques for the analyses of *geo-referenced* data.
- Aplications in diverses areas like atmospheric sciences, climatology, ecology, real estate, etc.
- **Example:** Study temperature and precipitation through,
    - daily measurements from *monitoring stations*;
    - a *mean surface* that considers changes in elevation or trends by *location*.

General Concepts

Interested on summaries of the data, but also in:

- *modeling* trends and correlations,
- *estimating* underlying parameters,
- *comparing* diverse model,
- *prediction* of observations at non-observed sites.

Some notation

- The basic format for geostatistical data is

$$(x_i, y_i); i = 1, 2, \ldots, n.$$

- x_i represents some spatial location (2-D, 3-D).
- y_i is a response variable corresponding to x_i.
- The values of x_i belong to a domain A.
- The $x_i's$ could define fixed *stations*, a grid on A or points generated by a *stochastic* process.

Example: Elevations data

- y_i elevation on a surface with 52 locations ($x_i' s$) on a square A (side 6.7 units).
- Use **R** packages, **geoR** and **geoRglm** from http://www.r-project.org.
- In Windows go to "Packages", "Install" and search **geoR**

```
require(geoR)
data(elevation)
help(elevation)
points(elevation, cex.min=2,cex.max=2,col="gray")
summary(elevation)
plot(elevation, lowess=T)
```

Example: Elevations data

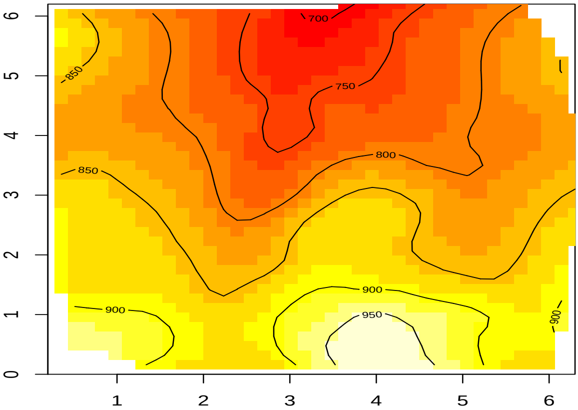
Goal: Build a *continuous* map of elevations on *A*.

- Useful to produce images and contours.
- Download and call **akima**: library(akima).

```
library(akima)
int=interp.new(elevation$coords[,1],elevation$coords[,2]
,elevation$data)
image(int,xlim=range(elevation$coords[,1]),
ylim=range(elevation$coords[,2]))
contour(int,add=T)
```

Akima provides a linear interpolation. Does not permit formal predictions.

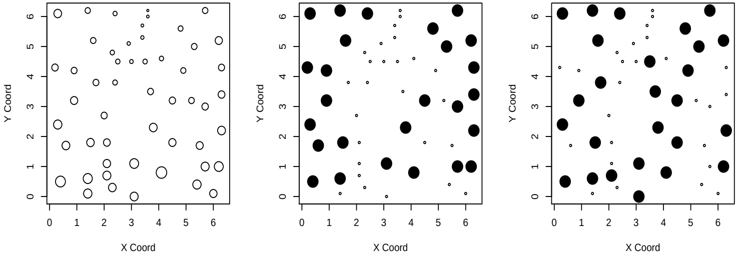
Interpolation with akima



Exploratory analysis of the data

```
par(mfrow=c(1,3))
points(elevation, cex.max=2.5)
points(elevation,trend="1st",pt.div=2,abs=T,cex.max=2.5)
points(elevation,trend="2nd",pt.div=2,abs=T,cex.max=2.5)
```

- Size proportional to magnitude of observations.
- Not simple to visualize the impact on spatial correlations.

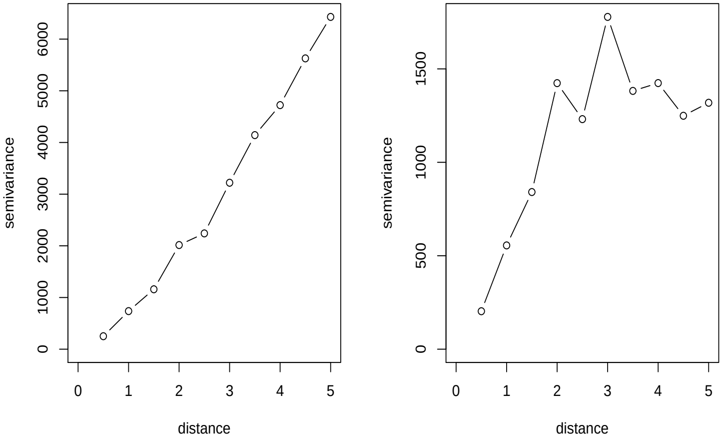

Y Coord
X Coord
0
1
2
3
4
5
6

Empirical Variogram

- Measure to study spatial correlations in terms of *distance*.
- For a set $(x_i, y_i), i = 1, 2, \ldots, n$, the empirical semi-variogram ordinates: $v_{ij} = \frac{1}{2}(y_i - y_j)^2$.
- If $y_i's$ have a constant mean and variance, $E(v_{ij}) = \sigma^2\{1 - \rho(x_i, x_j)\}$ where $\rho(x_i, x_j)$ is the correlation between y_i and y_j; σ^2 is the variance of y_i.
- If y_i arises from a *stationary* process, $\rho(x_i, x_j)$ depends on the distance between x_i and x_j.
- $E(v_{ij}) \approx \sigma^2$, if $u_{ij} = ||x_i - x_j|| \to \infty$.

- The graph of v_{ij} versus the distances u_{ij} defines the variogram.
- For better interpretability, the $v_{ij}'s$ are averaged over a 'band' of distances.

```
plot(variog(elevation,uvec=seq(0,5,by=0.5)),type="b")
res1.v <- variog(elevation,trend="1st",
uvec=seq(0,5,by=0.5))
plot(res1.v,type="b")
```


semivariance
0
1000
2000
3000
4000
5000
6000
0
1
2
3
4
5
distance
semivariance
0
500
1000
1500
0
1
2
3
4
5
distance

Statistical Model

- Defined as: $Y_i = S(x_i) + Z_i; i = 1, 2, \ldots, n$.
  - The $Z_i's$ are mutually independent $N(0, \tau^2)$.
  - $S(x_i)$ its a realization of a *Gaussian Process*.
- A Gaussian process is such that the joint distribution of $(S_1, S_2, \ldots, S_k)$ follows a *Multivariate Normal* with some mean y covariance.
- $S(x)$ is a *Gaussian Stationary process* of mean μ, variance $\sigma^2 = Var(S(x))$ and correlation $\rho(u) = Corr(S(x), S(x'))$ where $u = ||x - x'||$.
- τ^2 is a measure error and represents "micro-scale" errors.

Statistical model

- Selection of $\rho(u)$ requires a carefull exploration of the variogram.
- $\rho(u)$ can adopt a parametric form (Matérn class):

$$\rho(u;\phi,\kappa) = \{2^{\kappa-1}\Gamma(\kappa)\}^{-1}(u/\phi)^{\kappa}K_{\kappa}(u/\phi)$$

- $K_{\kappa}(\cdot)$ is the modified *Bessel* function of order κ.
- ϕ is a range parameter.
- Determines the rate in which the correlation decays to 0.

Estimation

- The parameters, τ^2, μ, σ^2, κ and ϕ are unknown.
- Estimation via numerical *Maximum Likelihood*.
- Initial values for σ^2, ϕ are needed.

```
m10=likfit(elevation, ini=c(3000,2), cov.model="matern"
,kappa=1.5)
m10
likfit: estimated model parameters:
      beta    tausq     sigmasq         phi
" 848.316"  "48.138" "3510.233"   "1.198"
Practical Range with cor=0.05 for
asymptotic range: 5.685412
likfit: maximised log-likelihood = -242.1
```

- $\hat{\mu} = 848.3$, $\hat{\tau^2} = 48.1$, $\hat{\sigma^2} = 3510.2$, $\hat{\phi} = 1.2$
- The exploratory analysis of the data suggests a model,

$$\mu(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

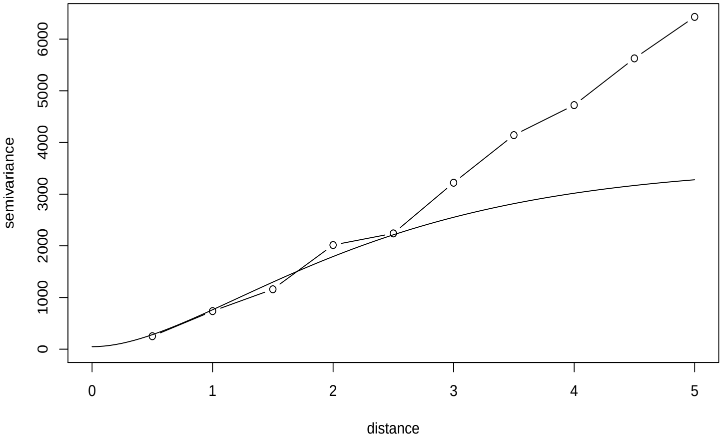
```
m11=likfit(elevation,trend="1st",ini=c(1300,2),
 cov.model="matern",kappa=1.5)
m11
likfit: estimated model parameters:
   beta0    beta1    beta2     tausq  sigmasq    phi
" 912.48" "-4.99" "-16.46" "34.89" "1693.13" "0.81"
Practical Range with cor=0.05 for asymptotic
range: 3.824244
likfit: maximised log-likelihood = -240.1
```

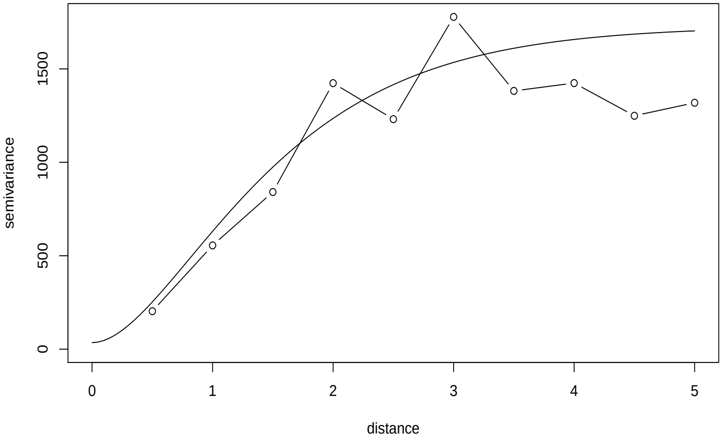
Comparison of Variograms

- To evaluate fit, compare empirical variogram to the theoretically estimated variogram.
- For a stationary process $Y_i = S(x_i) + Z_i$, the variogram is:

$$V_Y(u) = \frac{1}{2} Var\{S(x) - S(x-u)\} = \tau^2 + \sigma^2\{1 - \rho(u)\}$$

```
plot(variog(elevation,uvec=seq(0,5,by=0.5)),
type="b")
lines.variomodel(seq(0,5,by=0.5),
cov.pars=c(m10$sigmasq,m10$phi),
cov.model="mat",kap=1.5,nug=m10$tausq)
```


semivariance
0
1000
2000
3000
4000
5000
6000
0
1
2
3
4
5
distance


semivariance
0
500
1000
1500
0
1
2
3
4
5
distance

Spatial Prediction

- Also known as *Kriging*.
  - Prediction with minimum mean square error.
  - Estimated parameters are treated as *true*.
- For observations $y = (y_1, \ldots, y_n)$ and a spatial point x, the predictor $\hat{S}(x)$ is such that minimizes

$$E(\hat{S}(x) - S(x))^2$$

Spatial Prediction

- Then, $\hat{S}(x) = E(S(x)|y)$ and so

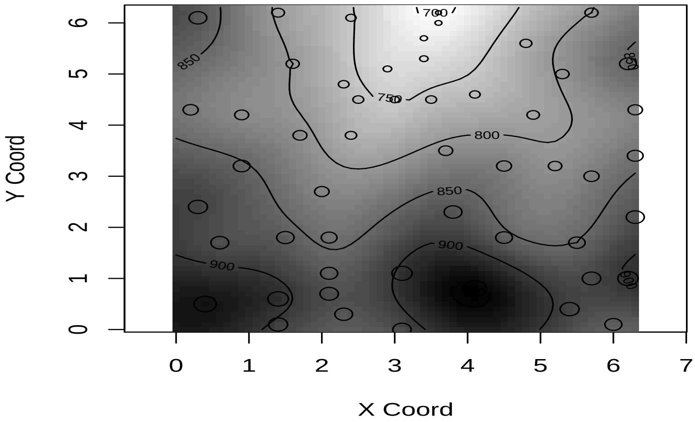
$$\hat{S}(x) = \mu + \sum_{i=1}^{n} w_i(x)(y_i - \mu)$$

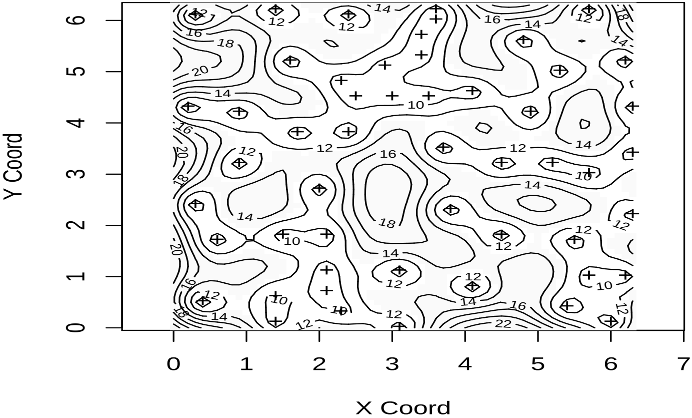
- $w_i(x)$ depend on σ^2, τ^2 and ϕ.
- Considers the standard error of prediction

$$SE(x) = \sqrt{Var\{S(x)|y\}}$$

Prediction in GeoR

```
locs=pred_grid(c(0,6.3),c(0,6.3),by=0.1)
KC=krige.control(type="sk",obj.mod=m10)
sk=krige.conv(elevation, krige=KC,loc=locs)
# spatial maps of predictions
pred.lim=range(c(sk$pred,skt$pred)
sd.lim=range(c(sk$kr,skt$kr))
image(sk,col=gray(seq(1,0,l=51)),zlim=pred.lim)
contour(sk,add=T,nlev=6)
points(elevation,add=TRUE,cex.max=2)
```


700
750
800
850
850
900
900
Y Coord
X Coord
0
1
2
3
4
5
6
7


Y Coord
X Coord
0
1
2
3
4
5
6
7

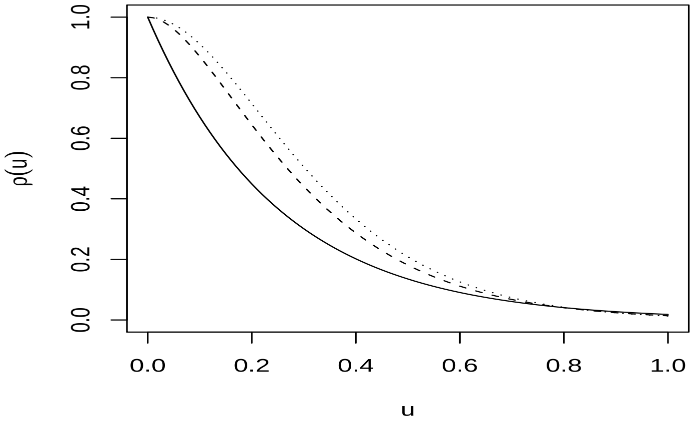
Correlation Functions

- Matérn Family:

$$\rho(u;\phi,\kappa) = \{2^{\kappa-1}\Gamma(\kappa)\}^{-1}(u/\phi)^{\kappa}K_{\kappa}(u/\phi)$$

- If $\kappa = 0.5$, $\rho(u) = exp(-u/\phi)$, Exponential model (continuous, non diferentiable at 0).
- If $\kappa \to \infty$, $\rho(u) \to exp\{-(u/\phi)^2\}$, Gaussian model (infinitely diferentiable).
- *Practical Range :* u value where $\rho(u) = 0.05$.

```
x=seq(0,1,l=101)
plot(x,cov.spatial(x,cov.model="mat",kappa=0.5,
cov.pars=c(1,0.25)),type="l")
```

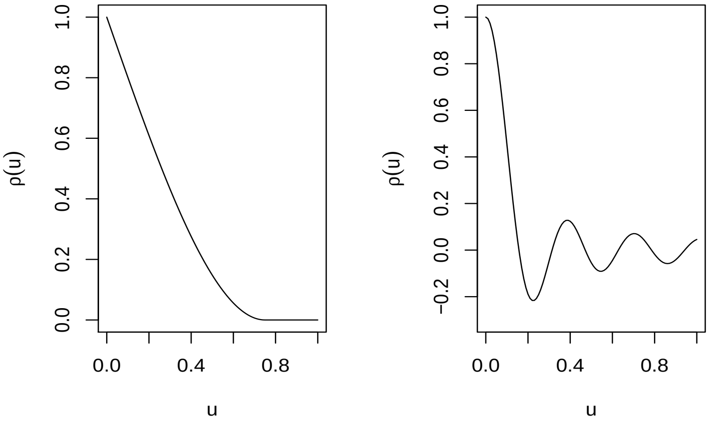


ρ(u)
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
u

Other families

- Power Exponential: $\rho(u) = exp(-(u/\phi)^\kappa)$, $0 < \kappa \leq 2$ (monotone decreasing).
- Spherical: $\rho(u) = 1 - \frac{3}{2}(u/\phi) + \frac{1}{2}(u/\phi)^3; 0 \leq u \leq \phi$; $\rho(u) = 0; u > \phi$.
- Non-monotone: $\rho(u) = (u/\phi)^{-1} sin(u/\phi); \phi > 0$

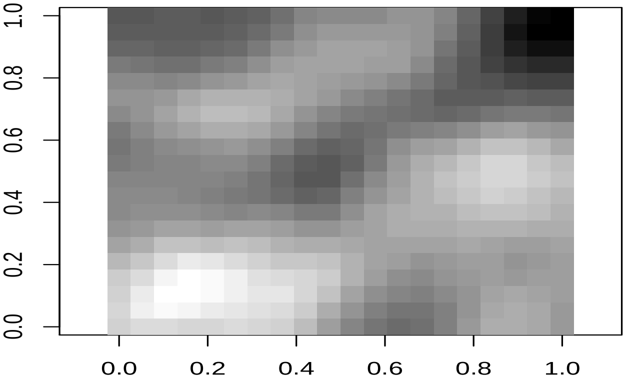
```
x=seq(0,1,l=101)
plot(x,cov.spatial(x,cov.model="spherical",
cov.pars=c(1,0.75)),type="l")
plot(x,cov.spatial(x,cov.model="wave",
cov.pars=c(1,0.05)),type="l")
```


ρ(u)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.4
0.8
u
ρ(u)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.4
0.8
u

Simulation of a Gaussian Process

- Could be done through the R-function *grf*.
- Could use a regular *grid* or random points.
- One specifies the model and its parameters.

```
image(grf(20^2,grid="reg",cov.pars=c(1,0.13),
cov.model="mat",kappa=2.5),
col=gray(seq(1,0,l=51)),xlab=" ", ylab=" ")
```



0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

Bayesian Analysis in Spatial Statistics

- Takes into account *uncertainty* in parameters.
- Does not solely depend on a point estimator $\hat{\theta}$.
- Elegant solution that requires Monte Carlo simulation (MCMC) and priors.
- If S is the spatial process; Y is the data:

$$p(Y, S, \theta) = p(\theta)p(S|\theta)p(Y|S, \theta)$$

- $p(\theta)$ represents a prior distribution on θ.
- The predictive distribution,

$$p(S|Y) = \int p(S|Y, \theta)p(\theta|Y)d\theta.$$

- Conditional predictions are 'integrated' respect to the posterior $p(\theta|Y)$.

The Basic Model

- Gaussian model with spatial 'trend':

$$Y \sim N(D\beta, \sigma^2 R(\phi) + \tau^2 I)$$

  - $D\beta$ is a regression term.
  - σ^2 is a spatial variance; τ^2 correspond to measurement error or micro-scale variability.
  - $R_{ij} = \rho(||x_i - x_j||, \phi)$ with $\rho(\cdot)$ a correlation function.
- If $\theta = (\beta, \sigma^2, \tau^2, \phi)$, we need a prior $p(\theta)$, so the *posterior* is:

$$p(\theta|y) \propto f(y|\theta)p(\theta)$$

  equivalent to $p(\beta, \sigma^2, \tau^2, \phi|Y)$.

- Its typical to assume the parameters are *independent*:

$$p(\theta) = p(\beta)p(\sigma^2)p(\tau^2)p(\phi)$$

- Its common to adopt a *Multivariate Normal* on β, an *Inverse Gamma* on σ^2 and τ^2.
- For $p(\phi)$ its usual to adopt a *Uniform* or *Inverse Gamma* prior.
- Non-informative: $p(\beta) \propto 1$, however ϕ and at least one of σ^2 or τ^2 must have an informative prior.
- Model requires use of *Metropolis-Hastings* or *Slice Sampler*.

Alternative in geoR

- Prior of the form

$$p(\beta, \sigma^2 | \phi, \nu^2) p(\phi, \nu^2)$$

  with $\nu^2 = \tau^2/\sigma^2$, $p(\phi, \nu^2)$ a discrete distribution over a reasonable range.
- The posterior distribution has a structure,

$$p(\beta, \sigma^2, \phi, \nu^2 | y) = p(\beta, \sigma^2 | y, \phi, \nu^2) p(\phi, \nu^2 | y)$$

- The parameters are simulated iteratively.

- For a prediction of $S^* = S(x^*)$ where x^* is an arbitrary site,

$$p(S^*|y) = \int p(S^*|\theta, y)p(\theta|y)d\theta.$$

- Method of composition: For each sample of $p(\theta|y)$, one simulates a sample from $p(S^*|\theta, y)$.

$$\theta_1, \theta_2, \ldots, \theta_m \rightarrow S_1^*, S_2^*, \ldots, S_m^*$$

- The function *krige.bayes( )* implements Bayesian inference on the Gaussian model.

```
args(krige.bayes)
function(geodata, coords = geodata$coords,
     data = geodata$data,
    locations = "no", borders, model,prior,output)
NULL
```

- Its use is more simple through *model.control*, *prior.control* and *output.control*.
- A trend is specified with *trend.d* and *trend.l* for prediction.
- A discrete uniform is specified for ϕ and $\nu^2 = \tau^2/\sigma^2$.
- For β and σ^2, $p(\beta, \sigma^2) \propto 1/\sigma^2$.

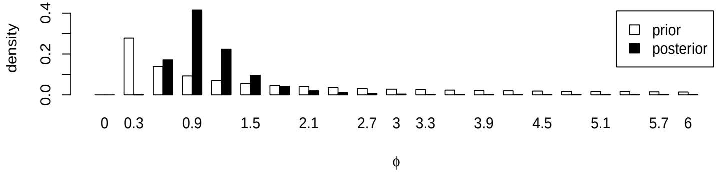
Elevation data

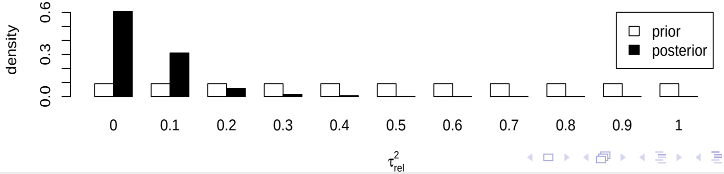
```
MC=model.control(trend.d="1st",trend.l
="1st",kappa=1.5)
PC=prior.control(phi.discrete=seq(0,6,l=21),
phi.prior="reciprocal",
tausq.rel.prior="unif",tausq.rel.discrete
=seq(0,1,l=11))
OC=output.control(n.post=1000,moments=T)
```

Now *krige.bayes* can be used:

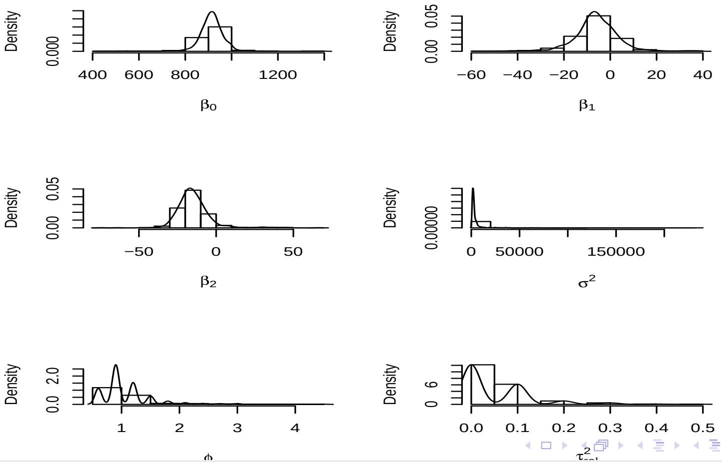
```
set.seed(268)
locs=pred_grid(c(0,6.3),c(0,6.3),by=0.3)
skb = krige.bayes(elevation,loc=locs,model=MC,
      prior=PC,output=OC)
par(mfrow=c(2,1));plot(skb)
par(mfrow=c(3,2));hist(skb)
```

Posterior distributions of ϕ and τ^2.



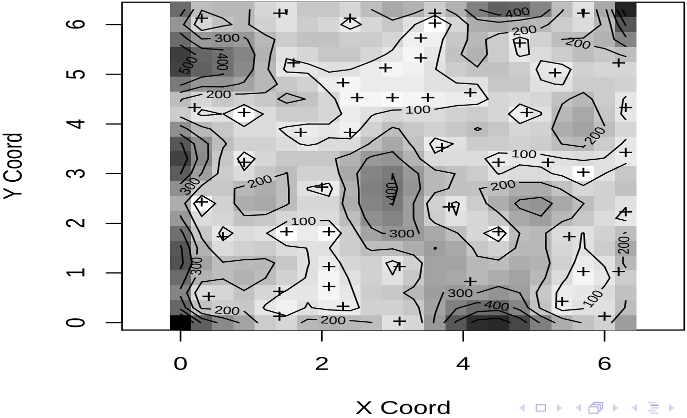


Posterior distribution on parameters

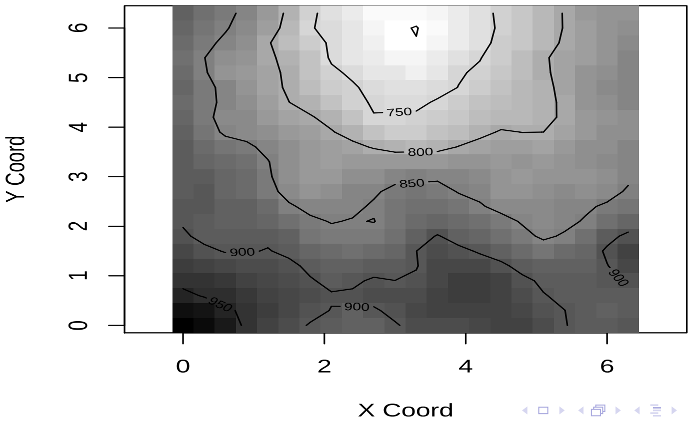


```
# Map of the mean of the predictive dist.
image(skb,col=gray(seq(1,0,l=51)))
points(elevation,add=TRUE,cex.max=2)
# Map of the variance of the predictive dist.
image(skb,val="variance",col=gray(seq(1,0,l=51)),
main="prediccion var")
contour(skb,val="variance",add=T)
points(elevation$coords,pch="+")
# Sample of the predictive dist.
image(skb,val="simulation",col=gray(seq(1,0,l=51)),
number.col=1,main="simulacion")
contour(skb,val="simulation",number.col=1,add=T)
```

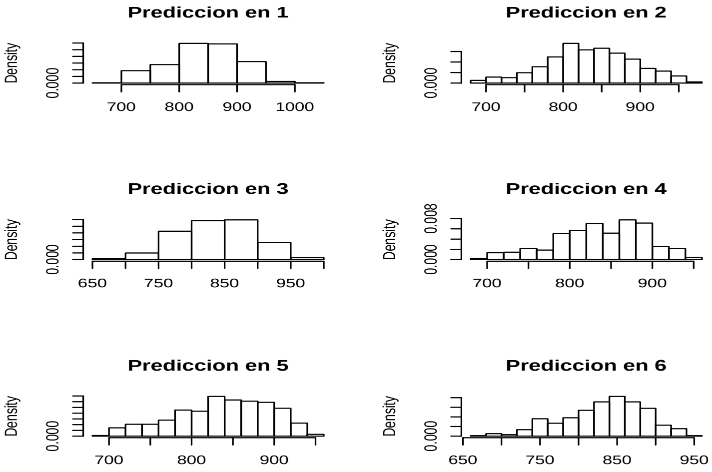
Map of the variance. Predictive distribution.



Map of one sample. Predictive distribution.



```
# Predictions for some points
names(skb$predictive)
dim(skb$predictive$simulations)
pred=skb$predictive$simulations
par(mfrow=c(3,2))
for(i in 1:6){hist(pred[,i],prob=T,xlab=" ",
main=paste("Prediction on",i)) }
names(skb$predictive)
```



Prediccion en 1
Density
0.000
700
800
900
1000
Prediccion en 2
Density
0.000
700
800
900
Prediccion en 3
Density
0.000
650
750
850
950
Prediccion en 4
Density
0.000
0.008
700
800
900
Prediccion en 5
Density
0.000
700
800
900
Prediccion en 6
Density
0.000
650
750
850
950

A generalized linear model (Poisson)

- The observations $Y_i, i = 1, \ldots, n$ are counts on x_i.
- The $Y_i's$ are conditionally independent, $Y_i \sim Poisson(\mu_i)$.

$$log(\mu_i) = \alpha + S(x_i) + Z_i.$$

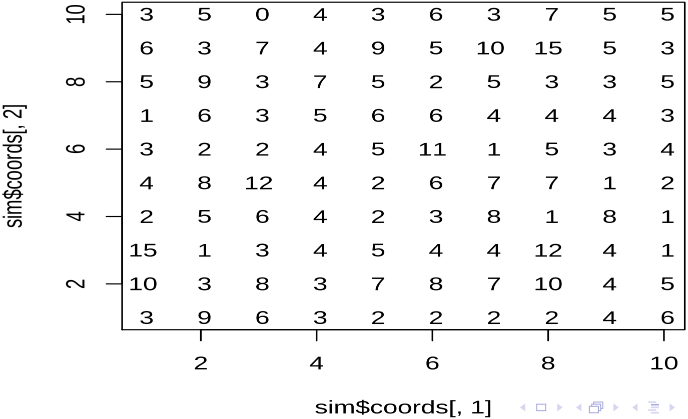
  - $S(\cdot)$ is a *Gaussian Process* with variance σ^2 and correlation $\rho(u)$
  - The $Z_i's$ are independent Normal $N(0, \tau^2)$.
- Induces spatial dependency through the parameters.
- Binomial case: $log(p(x_i)/(1 - p(x_i))) = \alpha + S(x_i)$.

Simulation of the generalized spatial process

- Through the function *grf* that simulates a Gaussian process plus noise.

```
sim <- grf(grid =expand.grid(x=seq(1,10,l = 10),
  y =seq(1, 10, l = 10)),
  cov.pars = c(0.1, 0.2))
  attr(sim,"class") <- "geodata"
  sim$units.m <- c(rep(5, 100))
  sim$data <- rpois(100, lambda =
  sim$units.m*exp(sim$data))
plot(sim$coords[,1],sim$coords[,2],type = "n")
text(sim$coords[,1],sim$coords[,2],format(sim$data))
```

Simulation of Poisson process



Estimation and Predictions

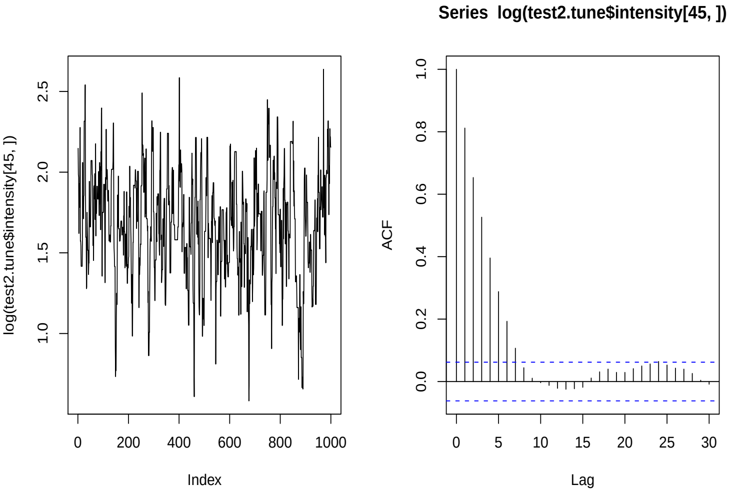
- Use *pois.krige* inside of *geoRglm*.
- Through *conditional simulation* with Gaussian process parameters fixed.
- Look for a 60% acceptance rate.

```
model2=krige.glm.control(cov.pars =c(1,1),beta = 1)
test2.tune=pois.krige(p50,krige=model2,
mcmc.input=list(S.scale=0.2,thin = 1))
```

- The optimal value is around $S.scale = 0.5$.

- Its necessary to study how the chain *mixes*.

```
test2.tune=pois.krige(p50, krige = model2, mcmc.input
      = list(S.scale = 0.5, thin = 1))
par(mfrow=c(1,2))
plot(log(test2.tune$intensity[45,]), type = "l")
acf(log(test2.tune$intensity[45,]), type =
"correlation", plot = TRUE)
```



Series log(test2.tune$intensity[45,])
log(test2.tune$intensity[45,])
1.0
1.5
2.0
2.5
0
200
400
600
800
1000
Index
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
30
Lag

- Predictions of intensity at 2 points, (0.5,0.5) and (1,0.4).

```
test2=pois.krige(p50,locations=
 cbind(c(0.5,0.5),c(1,0.4)),
 krige=model2, mcmc.input
 =mcmc.control(S.scale = 0.5),
 output=output.glm.control(sim.predict = TRUE))
```

- Output includes predictions, variances of predictions and Monte Carlo errors.

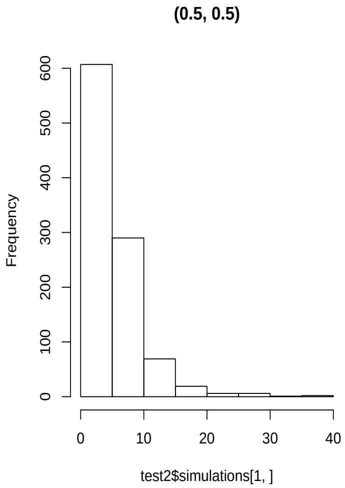
```
test2$predict
test2$krige.var
test2$mcmc.error
```

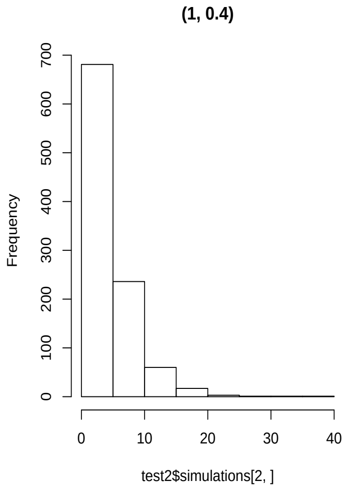
- The predictions of the intensity at 2 points are in test2$simulations.

```
par(mfrow = c(1,2))
hist(test2$simulations[1,],main="(0.5, 0.5)")
hist(test2$simulations[2,],main ="(1, 0.4)")
```

- For "beta" to follow a uniform distribution

```
model2.u=krige.glm.control(cov.pars = c(1,1),
beta = 1,type.krige = "ok")
test2.unif.beta =pois.krige(p50,krige =model2.u,
mcmc.input = list(S.scale = 0.5))
```


(0.5, 0.5)
Frequency
0
100
200
300
400
500
600
0
10
20
30
40
test2$simulations[1,]


(1, 0.4)
Frequency
0
100
200
300
400
500
600
700
0
10
20
30
40
test2$simulations[2,]

Full Bayes Analysis

- Implemented in the function pois.krige.bayes

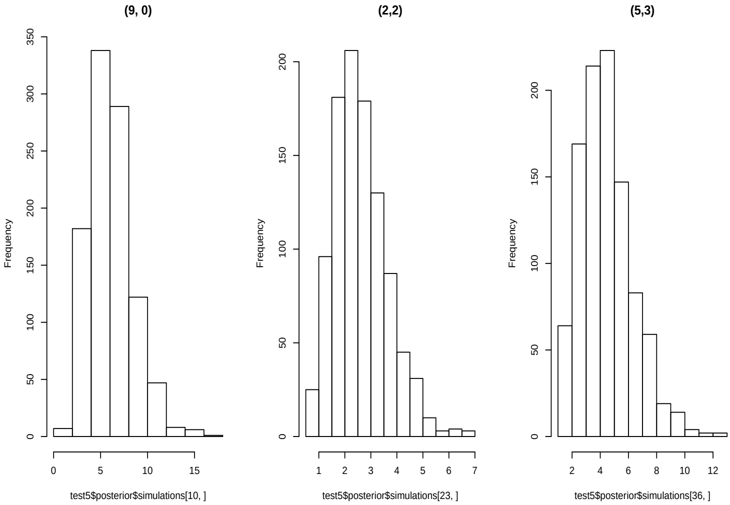
```
prior5=prior.glm.control(phi.prior="fixed", phi = 0.1)
mcmc5.tune=mcmc.control(S.scale = 0.01, thin = 1)
test5.tune=pois.krige.bayes(p50, prior =
 prior5, mcmc.input = mcmc5.tune)
```

- S.scale parameter adjusted to improve acceptance rate.

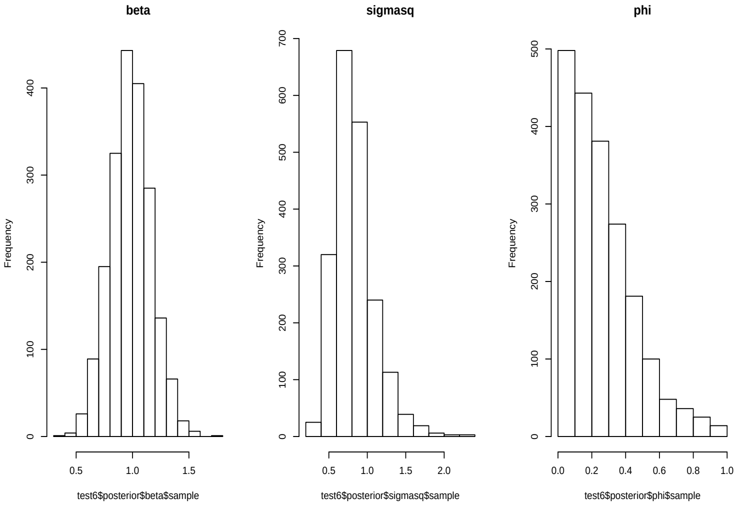
```
mcmc5=mcmc.control(S.scale = 0.075, thin = 100)
test5=pois.krige.bayes(p50, locations =
t(cbind(c(2.5,3),c(-6050,-3270))),
 prior = prior5, mcmc.input = mcmc5,
output = list(threshold=10,
quantile = c(0.05,0.99)))
```

- "Posterior" contains information about the posterior distribution of parameters and simulations at data points.
- "Predictive" contains information about predictions.
- "threshold=10" gives the probability that predictions are lower than 10.
- "quantiles" gives 0.05 and 0.99 quantiles of predictive (test5$predictive, test5$quantiles).

```
par(mfrow = c(1,3))
hist(test5$posterior$simulations[10,],main="(9, 0)")
hist(test5$posterior$simulations[23,],main= "(2,2)")
hist(test5$posterior$simulations[36,],main="(5,3)")
```


(9, 0)
(2,2)
(5,3)
Frequency
test5$posterior$simulations[10,]
test5$posterior$simulations[23,]
test5$posterior$simulations[36,]

```
# Example with uncertainty in other parameters
mcmc6.tune<-mcmc.control(S.scale=0.075,n.iter=2000,
thin=100,phi.scale = 0.01)
prior6<-prior.glm.control(phi.prior="uniform",
phi.discrete=seq(0.02,1, 0.02),tausq.rel = 0.05)
test6.tune <- pois.krige.bayes(p50,prior =
prior6,mcmc.input= mcmc6.tune)
#This may take some time.
mcmc6 <-mcmc.control(S.scale=0.075,
n.iter=400000,thin=200,burn.in=5000,
phi.scale=0.12,phi.start=0.5)
test6 <- pois.krige.bayes(p50,locations =
t(cbind(c(2.5,3.5),c(-60,-37))),
prior=prior6,mcmc.input=mcmc6)
# some posterior distributions.
par(mfrow=c(1,3))
hist(test6$posterior$beta$sample,main="beta")
hist(test6$posterior$sigmasq$sample,main="sigmasq")
```


beta
sigmasq
phi
Frequency
test6$posterior$beta$sample
test6$posterior$sigmasq$sample
test6$posterior$phi$sample
0
100
200
300
400
500
600
700
0.5
1.0
1.5
2.0
0.0
0.2
0.4
0.6
0.8

References

- Diggle P.J. and Ribeiro P. J. (2007) *Model-based Geostatistics*, Springer-Verlag, New York.
- Banerjee, S, Carlin B. and Gelfand A. (2004) Hierarchical Modeling and Analysis for Spatial Data. Chapman and Hall/CRC