Model-based Geostatistics: geospatial statistical methods for public health applications

Peter J Diggle and Emanuele Giorgi

CHICAS, Lancaster University

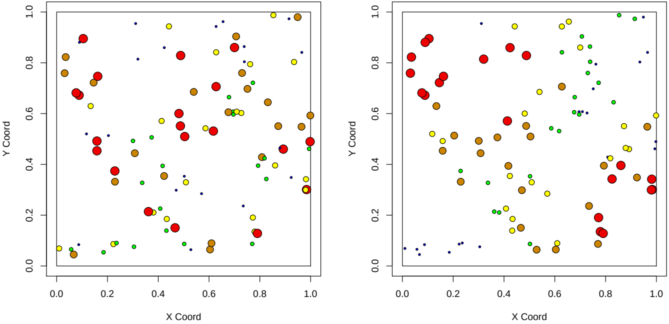
Linear geostatistical models

The first law of geography

All things are related, but close things are more strongly related than distant things

Completely random variation: measurements at different locations are statistically independent

Spatial variation: measurements at different locations are statistically dependent, and the strength of this dependence varies according to their relative locations



Notation

- $\mathbf{Y} = \{\mathbf{Y_i} : \mathbf{i} = 1, \ldots, \mathbf{n}\}$ is the measurement data
- $\mathcal{X} = \{\mathbf{x_i} : \mathbf{i} = 1, \ldots, \mathbf{n}\}$ is the sampling design
- **A** is the region of interest
- $\mathbf{Y}^* = \{\mathbf{Y}(\mathbf{x}) : \mathbf{x} \in \mathbf{A}\}$ is the measurement process
- $\mathbf{S}^* = \{\mathbf{S}(\mathbf{x}) : \mathbf{x} \in \mathbf{A}\}$ is the signal process
- $\mathbf{S} = \{\mathbf{S}(\mathbf{x_i}) : \mathbf{i} = 1, \ldots, \mathbf{n}\}$
- $\mathbf{T} = \mathcal{F}(\mathbf{S}^*)$ is the target for prediction
- $[\mathbf{S}^*, \mathbf{Y}] = [\mathbf{S}^*][\mathbf{Y}|\mathbf{S}^*]$ is the geostatistical model

The linear Gaussian model

Model:

- **Stationary Gaussian process $S(x) : x \in \mathbb{R}^2$**
  - $\mathrm{E}[S(x)] = \mu$
  - $\mathrm{Cov}\{S(x), S(x')\} = \sigma^2 \rho(\|x - x'\|)$
- **Mutually independent $Y_i | S(\cdot) \sim \mathrm{N}(S(x_i), \tau^2)$**

Questions:

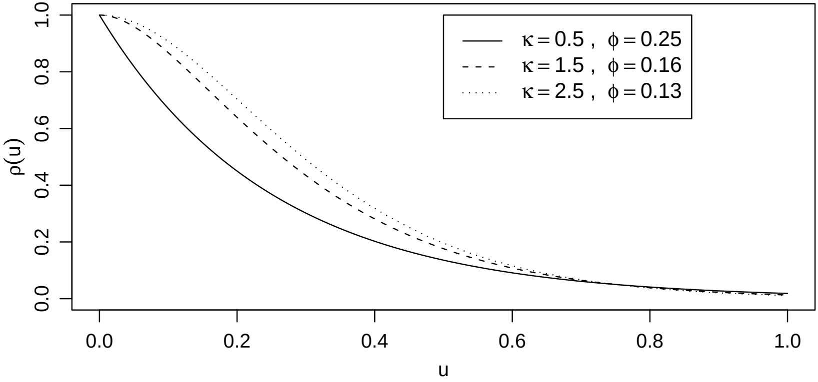
- **covariates? $\mu \to \mu(x) = d(x)'\beta$**
- **how to specify the correlation function $\rho(u)$?**

The Matérn family of correlation functions

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

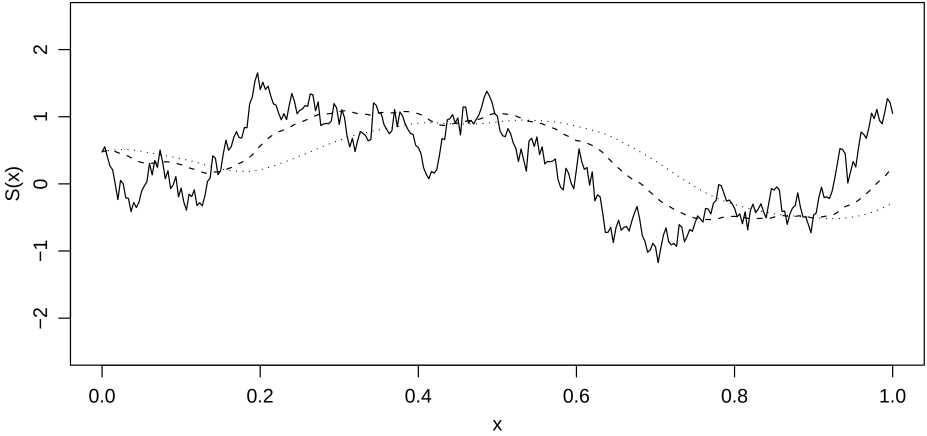
- **parameters $\kappa > 0$ and $\phi > 0$**
- **$K_{\kappa}(\cdot)$: modified Bessel function of order κ**
- **$\kappa = 0.5$ gives $\rho(u) = \exp\{-u/\phi\}$**
- **$\kappa \to \infty$ gives $\rho(u) = \exp\{-(u/\phi)^2\}$**
- **κ and ϕ are not orthogonal:**
  - **helpful re-parametrisation: $\phi \to \alpha = 2\phi\sqrt{\kappa}$**
  - **but estimation of κ is difficult**

The Matérn correlation function



- $\boldsymbol{\kappa \leq 1 \Rightarrow}$ **S(x) is continuous but non-differentiable**
- $\boldsymbol{\kappa >}$ **c** $\boldsymbol{\Rightarrow}$ **S(x) is c times differentiable**

Matérn simulated realisations



Parameter estimation using the variogram

What not to do and how to do it

- **weighted least squares criterion:**

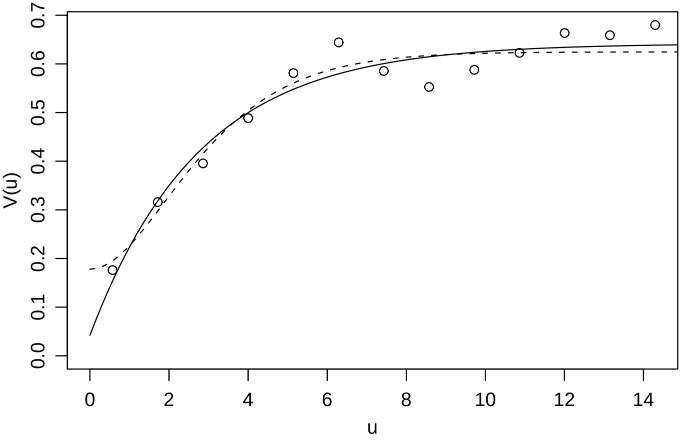
$$W(\theta) = \sum_k n_k \{\hat{V}(u_k) - V(u_k;\theta)\}^2$$

  where θ denotes vector of covariance parameters

- **arbitrary upper limit for u_k**
- **false analogy with regression modelling of independently replicated data**

Comments on variogram fitting

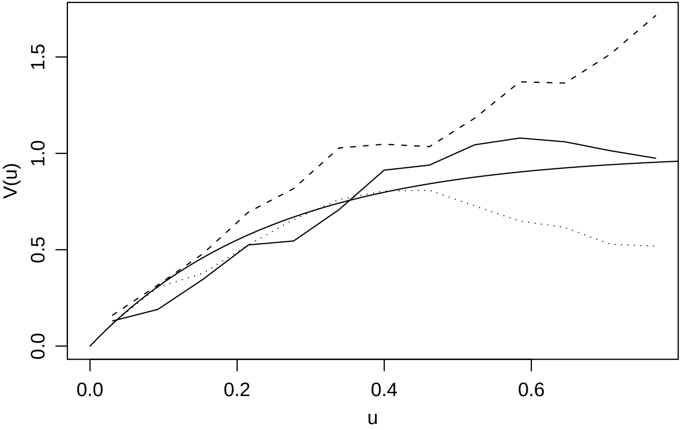
Different extrapolations at origin give equally good fits



Comments on variogram fitting (2)

Correlation between variogram points induces smoothness, giving false impression of precision

Three simulations from the same model.



Parameter estimation: maximum likelihood

$$\mathbf{Y} \sim \mathrm{MVN}(\mu \mathbf{1}, \sigma^2 \mathbf{R} + \tau^2 \mathbf{I})$$

- **R is $n \times n$ matrix, $(i, j)^{th}$ element $\rho(u_{ij})$**
- **$u_{ij} = ||x_i - x_j||$, Euclidean distance between x_i and x_j**

Adding explanatory variables is technically straightforward:

$$\mu(x_i) = d(x_i)'\beta$$

$$\mathbf{Y} \sim \mathrm{MVN}(\mathbf{D}\beta, \sigma^2 \mathbf{R} + \tau^2 \mathbf{I})$$

Gaussian log-likelihood function

$$\mathbf{Y} \sim \mathrm{MVN}(\mathbf{D}\beta, \sigma^2\mathbf{R} + \tau^2\mathbf{I})$$

- **write** $\nu^2 = \tau^2/\sigma^2$**, hence** $\sigma^2\mathbf{V} = \sigma^2(\mathbf{R} + \nu^2\mathbf{I})$
- **log-likelihood function is maximised for**

$$\hat{\beta}(\mathbf{V}) = (\mathbf{D}'\mathbf{V}^{-1}\mathbf{D})^{-1}\mathbf{D}'\mathbf{V}^{-1}\mathbf{y}$$

$$\hat{\sigma}^2 = \mathbf{n}^{-1}(\mathbf{y} - \mathbf{D}\hat{\beta})'\mathbf{V}^{-1}(\mathbf{y} - \mathbf{D}\hat{\beta})$$

- **substitute** $(\hat{\beta}, \hat{\sigma^2})$ **to give reduced maximisation problem**

$$\mathbf{L}^*(\nu^2, \phi, \kappa) \propto -0.5\{\mathbf{n}\log|\hat{\sigma}^2| + \log|(\mathbf{R} + \nu^2\mathbf{I})|\}$$

choosing κ **from a discrete set, e.g.** $\kappa = 0.5, 1.5, 2.5$

A philosophical problem and its resolution

- In a linear geostatistical model with explanatory variables, write $\mu(\mathrm{x}) = \mathrm{d}(\mathrm{x})'\beta$
- Then, the data are generated by the formula

$$\mathrm{d_i} = \mu(\mathrm{x_i}) + \mathrm{s}(\mathrm{x_i}) + \mathrm{z_i}$$

  where
  - $\mu(\mathrm{x})$ is a deterministic function of x
  - S(x) is a realisation of the stochastic process S(x)

Problem

- without independent replication of the spatial process, how can you distinguish between $\mu(\mathrm{x})$ and s(x)?

Resolution

- **empirical:** use $\mu(\mathrm{x})$ and s(x) to describe large-scale and small-scale spatial variation, respectively
- **theoretical:** use contextual knowledge to transfer variation from s(x) (unexplained) to $\mu(\mathrm{x})$ (explained)

geoR: maximum likelihood estimation

```
mlfit2<-likfit(elevation,trend="2nd",ini.cov.pars=c(1000,1),
    cov.model="matern",kappa=2)
mlfit2
```

Prediction

The answer to any prediction problem is a probability distribution

Peter McCullagh, FRS

- **T = any quantity of scientific interest**
- **Y = data that can tell us something about T.**

The predictive distribution of T is the conditional probability distribution of T given Y

Geostatistical prediction

Let $S^* = \{S(x_1^*), ..., S(x_M^*)\}$ for any set of locations $\{x_1^*, ..., x_M^*\}$

- **$Y \sim$ multivariate Normal**
- **for the Gaussian linear model $S^*|Y \sim$ multivariate Normal**
- **hence simulate samples of S^* conditional on Y**
- **corresponding $T^* = \mathcal{T}(S^*)$ are samples from predictive distribution of T**

Minimum mean square error prediction

Model

- $[S^*]$ = probability distribution of underlying spatial process
- $[Y|S^*]$ = probability distribution of data conditional on underlying spatial process
- Bayes' theorem then gives us the predictive distribution $[S^*|Y]$

Mean square error

- $\hat{T} = t(Y)$ is a point predictor
- $\text{MSE}(\hat{T}) = \mathrm{E}[(\hat{T} - T)^2]$ is the mean square error

Theorem

1. $\text{MSE}(\hat{T})$ takes its minimum value when $\hat{T} = \mathrm{E}(T|Y)$.
2. $\mathrm{Var}(T|Y)$ estimates the achieved mean square error

Simple and ordinary kriging

$$\mathbf{Y} \sim \mathrm{MVN}(\mu\mathbf{1}, \sigma^2\mathbf{V})$$

$$\mathbf{V} = \mathbf{R} + (\tau^2/\sigma^2) \qquad\qquad \mathbf{R_{ij}} = \rho(\|\mathbf{x_i} - \mathbf{x_j}\|)$$

Target for prediction is $\mathbf{T} = \mathbf{S}(\mathbf{x})$

Write $\mathbf{r} = (\mathbf{r_1}, ..., \mathbf{r_n})$ where

$$\mathbf{r_i} = \rho(\|\mathbf{x} - \mathbf{x_i}\|)$$

Standard results on multivariate Normal then give $[\mathbf{T}|\mathbf{Y}]$ as multivariate Gaussian with mean and variance

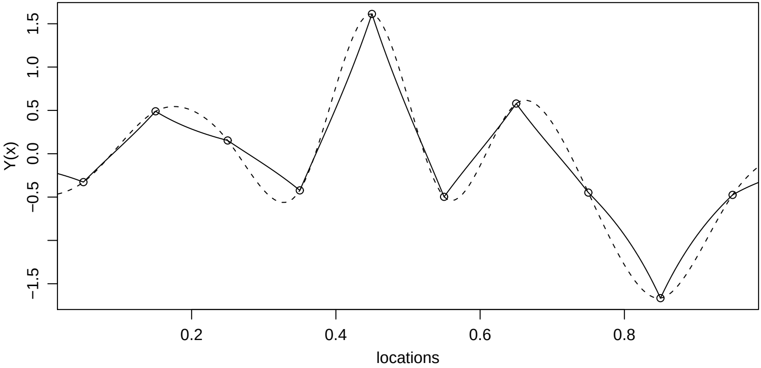
$$\hat{\mathbf{T}} = \mu + \mathbf{r}'\mathbf{V}^{-1}(\mathbf{Y} - \mu\mathbf{1})$$

$$\mathrm{Var}(\mathbf{T}|\mathbf{Y}) = \sigma^2(1 - \mathbf{r}'\mathbf{V}^{-1}\mathbf{r})$$

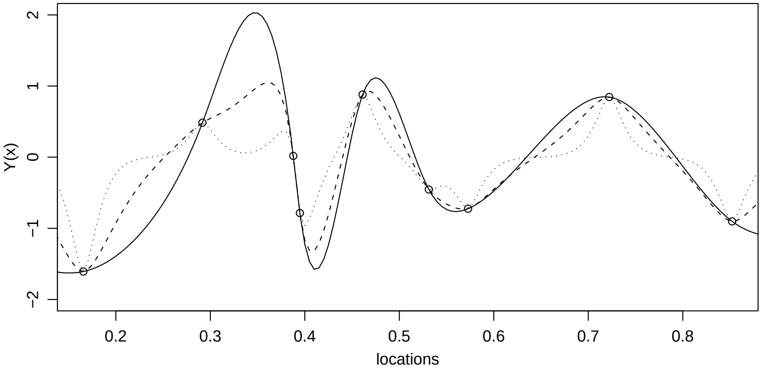
Simple kriging: $\hat{\mu} = \bar{\mathbf{Y}}$ **Ordinary kriging:** $\hat{\mu} = (\mathbf{1}'\mathbf{V}^{-1}\mathbf{1})^{-1}\mathbf{1}'\mathbf{V}^{-1}\mathbf{Y}$

Simple kriging: three examples

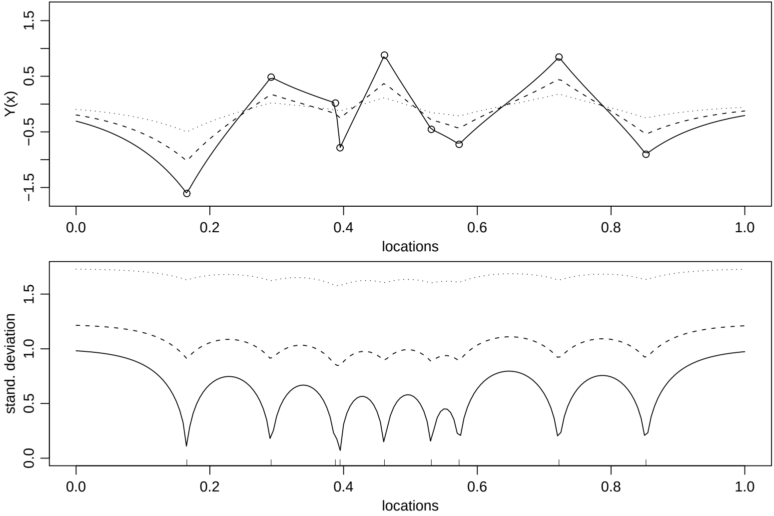
1. Varying κ (smoothness of S(x))



2. Varying ϕ (range of spatial correlation



3. Varying τ^2/σ^2 (noise-to-signal ratio)



Trans-Gaussian models

- **assume Gaussian model holds after point-wise transformation**
- **Box-Cox family is widely used**

$$Y_i^* = h_\lambda(Y_i) = \begin{cases} (Y_i^\lambda - 1)/\lambda & \text{if } \lambda \neq 0 \\ \log(Y_i) & \text{if } \lambda = 0 \end{cases}$$

Example: log-Gaussian kriging

- $T(x) = \exp\{S(x)\} \quad \hat{T}(x) = \exp\{\hat{S}(x) + v(x)/2\}$
- $S_1, \ldots, S_m$ **are a sample from** $[S|Y]$
- $T_i = \exp(S_i) \Rightarrow T_1, \ldots, T_m$ **are a sample from** $[T|Y]$

Reminder: Predicting non-linear functionals

- **minimum mean square error prediction is not invariant under non-linear transformation**
- **the complete answer to a prediction problem is the predictive distribution, $[T|Y]$**
- **Recommended strategy:**
  - **draw repeated samples from $[S^*|Y]$**
  - **calculate required summaries**

geoR: plug-in prediction

```
region<-matrix(c(0,0,6.5,0,6.5,6.5,0,6.5),4,2,T)
grid<-as.matrix(pred_grid(region,by=0.25))
KC<-krige.control(obj.model=mlfit2,trend.d="2nd",trend.l="2nd")
OC<-output.control(n.predictive=100)
set.seed(24367)
predictions<-krige.conv(geodata=elevation,locations=grid,
borders=region,krige=KC,output=OC)
image(predictions)
points(elevation,add=T)
par(mfrow=c(1,2))
hist(elevation$data,main="data")
predict.max<-NULL
for (sim in 1:100) {
   predict.max<-c(predict.max,max(predictions$simulations[,sim]))
   }
hist(predict.max,main="predicted maximum")
```

Bayesian inference

Model specification

$$[\mathbf{Y}, \theta] = [\theta][\mathbf{Y}|\theta]$$

- $[\mathbf{Y}|\theta]$ probability distribution of $\mathbf{Y}$ given parameter value θ
- $[\theta]$ prior probability distribution for θ
  (before you collect any data)

Parameter estimation

- Bayes' Theorem gives posterior distribution for θ
  (adding information from data)

$$[\theta|\mathbf{Y}] = [\mathbf{Y}|\theta][\theta]/[\mathbf{Y}]$$

where $[\mathbf{Y}] = \int [\mathbf{Y}|\theta][\theta] d\theta$

Bayesian inference for geostatistical models

Model specification

$$[\mathbf{Y}, \mathbf{S}, \theta] = [\theta][\mathbf{S}|\theta][\mathbf{Y}|\mathbf{S}, \theta]$$

- **[S] is an unobserved spatial stochastic process, representng the spatial phenomenon of scientifc interest**

Parameter estimation

- **integration gives likelihood function**

$$[\mathbf{Y}, \theta] = \int [\mathbf{Y}, \mathbf{S}, \theta] d\mathbf{S} = [\theta][\mathbf{Y}|\theta]$$

- **as before, Bayes' Theorem gives posterior distribution**

$$[\theta|\mathbf{Y}] = [\mathbf{Y}|\theta][\theta]/[\mathbf{Y}]$$

where $[\mathbf{Y}] = \int [\mathbf{Y}|\theta][\theta] d\theta$

Bayesian inference for geostatistical models (2)

Prediction

S denotes the spatial process of interest at data-locations

S^* denotes the same process at data and prediction locations

- **expand model specification to**

$$[Y, S^*, \theta] = [\theta][S|\theta][Y|S, \theta][S^*|S, \theta]$$

- **plug-in predictive distribution is**

$$[S^*|Y, \hat{\theta}]$$

- **Bayesian predictive distribution is**

$$[S^*|Y] = \int [S^*|Y, \theta][\theta|Y] d\theta$$

- **for any target $T = t(S^*)$, required predictive distribution $[T|Y]$ follows by direct calculation**

Notes

- **likelihood function is central to both classical and Bayesian inference**
- **Bayesian prediction is a weighted average of plug-in predictions, with different plug-in values of θ weighted according to their conditional probabilities given the observed data.**
- **Bayesian prediction is usually more conservative than plug-in prediction**

geoR: Bayesian estimation and prediction

```
MC<-model.control(trend.d="2nd",trend.l="2nd",kappa=2)
PC<-prior.control(beta.prior="flat",sigmasq.prior="sc.inv.chisq"
sigmasq=1000,df.sigmasq=4,phi.discrete=0.5*(1:5),
tausq.rel.prior="uniform",tausq.rel.discrete=0.1*(1:5))
OC<-output.control(n.posterior=100,n.predictive=100,
simulations.predictive=T,signal=T,moments=F)
set.seed(24367)
results.bayes<-krige.bayes(geodata=elevation,locations=grid,
borders=region,model=MC,prior=PC,output=OC)
```

geoR: plotting posterior distributions

```
plot(results.bayes)
posteriors.bayes<-results.bayes$posterior
posterior.sample<-posteriors.bayes$sample
par(mfrow=c(3,3))
for (i in 1:9){
   hist(posterior.sample[,i],main=" ")
   }
par(mfrow=c(1,1))
plot(posterior.sample[,2],posterior.sample[,3])
```

geoR: plotting predictive distributions

```
predictions.bayes<-results.bayes$predictive
image(unique(grid[,1]),unique(grid[,2]),
   matrix(predictions.bayes$mean.simulations,27,27))
points(elevation,add=T)
par(mfrow=c(1,2))
predict.max<-NULL

predict.max<-c(predict.max,max(predictions$simulations[,sim]))
for (sim in 1:100) {

predict.max<-c(predict.max,max(predictions$simulations[,sim]))
hist(predict.max,xlab="maximum", main="plug-in")
predict.bayes.max<-NULL
for (sim in 1:100) {
   predict.bayes.max<-c(predict.bayes.max,
      max(predictions.bayes$simulations[,sim]))
   }
hist(predict.bayes.max,xlab="maximum",main="Bayesian")
```