GeoModels Tutorial: analysis of spatio-temporal data with spatial locations changing over time using Gaussian random fields

Moreno Bevilacqua

Introduction

In this tutorial we show how to analyze geo-referenced spatio temporal data using Gaussian random fields (RFs) when the spatial coordinates change over time with the R package GeoModels (Bevilacqua and Morales-Oñate, 2018).

We first load the R libraries needed for the analysis and set the name of the model in the GeoModels package:

```
rm(list=ls())
require(devtools)
install_github("vmoprojs/GeoModels")
require(GeoModels)
require(fields)
model="Gaussian"  # model name in the  GeoModels package
set.seed(881)
```

Simulation of a space-time Gaussian random field with spatial coordinates changing over time

Let us consider a space-time Gaussian RF $Z = \{Z(\boldsymbol{s}, t), \boldsymbol{s} \in S, t \in B\}$, where $\boldsymbol{s}$ represents a location in the domain S and t represents a temporal instant the domain B. We assume that Z is stationary with zero mean, unit variance and correlation function given by $\rho(\boldsymbol{h}, u) = \mathrm{cor}(Z(\boldsymbol{s} + \boldsymbol{h}, t + u), Z(\boldsymbol{s}, t))$.

Then we consider the RF $Y = \{Y(\boldsymbol{s}, t), \boldsymbol{s} \in S, t \in T\}$ defined by the location and scale transformation:

$$Y(\boldsymbol{s}, t) = \mu(\boldsymbol{s}, t) + \sigma Z(\boldsymbol{s}, t) \tag{1}$$

where $\mu(\boldsymbol{s}, t) = X(\boldsymbol{s}, t)^T \boldsymbol{\beta}$ and $X(\boldsymbol{s}, t)$ is a k-dimensional vector of covariates and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)^T$ is a k-dimensional vector of (unknown) parameters (in this tutorial we fix $k = 2$). Then $\mathbb{E}(Y(\boldsymbol{s}), t) = X(\boldsymbol{s}, t)^T \boldsymbol{\beta}$, $\mathrm{var}(Y(\boldsymbol{s}, t)) = \sigma^2$ and $\mathrm{cov}(Y(\boldsymbol{s} + \boldsymbol{h}, t + u), Y(\boldsymbol{s}, t)) = \sigma^2 \rho(\boldsymbol{h}, u)$.

Suppose we want to simulate a realization of Y at $t_1 = 0, t_2 = 1, \ldots, t_T = 15$, $T = 16$ temporal instants and spatial locations (changing over time) $\boldsymbol{s}_{i(l)}$, with $i(l) = 1, \ldots, N_l$ and $l = 1, \ldots, T$, uniformly distributed in the unit square.

We first set the temporal instants and then the (changing over time) spatial coordinates with associated covariates.

```
coordt=seq(0,15,1) # temporal instants
coordx_dyn=list() # dynamical spatial coordinates
X=list() # dynamical spatial covariates
minN=150; maxN=250
for(k in 1:length(coordt))
{
 NN=sample(minN:maxN,size=1)
 x = runif(NN, 0, 1)
 y = runif(NN, 0, 1)
 coordx_dyn[[k]]=cbind(x,y)
 X[[k]]=cbind(rep(1,NN),runif(NN))
}
```

Note that the both the dynamical spatial coordinates and the covariates are saved as a list. The number of location sites $N_1, \ldots N_{16}$ for each temporal instants are given by

```
unlist(lapply(coordx_dyn,nrow))
[1] 227 227 189 162 250 167 214 179 221 200 192 247 188 222 181 200
```

and the total number of space-time locations is given by $\sum_{i=1}^{16} N_i = N$, in our example $N = 3266$.

The spatial coordinates for the first two temporal instants are depicted in Figure 1.

```
plot(coordx_dyn[[1]],pch=20,xlab = "",ylab="")
plot(coordx_dyn[[2]],pch=20,xlab = "",ylab="")
```

We then specify the mean, variance and nugget parameters

```
mean = 0.2; mean1= -0.3
sill = 1; nugget = 0
```

where `mean`, `mean1` and `sill` are respectively β_1, β_2 and σ^2.

For the correlation function we assume a simple spatially isotropic and symmetric in time double exponential model

$$\rho((\boldsymbol{h},u);\alpha_s,\alpha_t) = e^{-\frac{||\boldsymbol{h}||}{\alpha_s}-\frac{|u|}{\alpha_t}} \tag{2}$$

Then we set the name of the correlation model and the associated parameters and save all the parameters as a list:

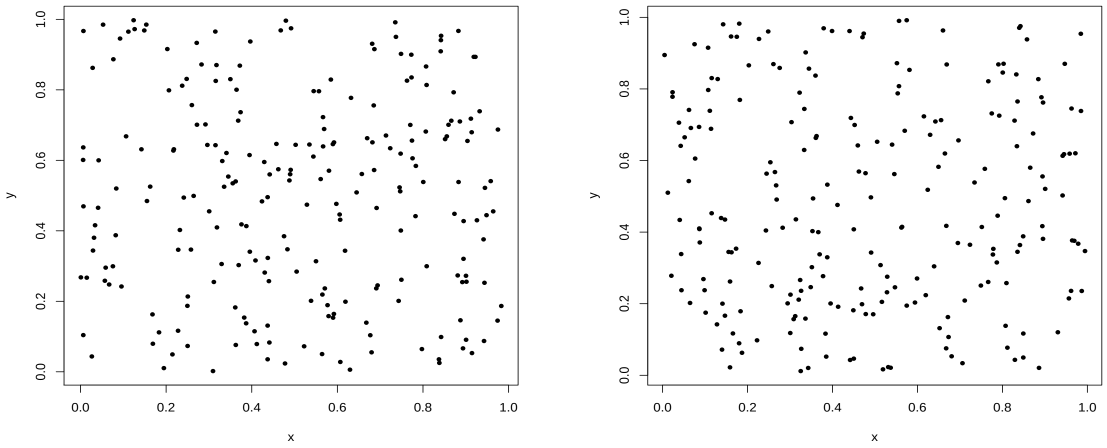

Figure 1: Spatial coordinates for the first two temporal instants

```
corrmodel = "Exp_Exp"
scale_s = 0.3/3
scale_t = 4/3
param = list(mean=mean,mean1=mean1,sill=sill,nugget=nugget,
              scale_s=scale_s,scale_t=scale_t)
```

We are now ready to simulate the space time Gaussian RF using the function `GeoSim`:

```
ss1 = GeoSim(coordx_dyn=coordx_dyn, coordt=coordt,
    corrmodel=corrmodel,X=X,model=model,param=param)$data
```

The simulation is performed using Cholesky decomposition. Note that the option `coordx_dyn` allows to specific dynamical spatial coordinates as a list.

Estimation of Gaussian space-time random fields

Let us assume that we observe the RF Y, at a finite set of spatial location sites changing over time *i.e.*, $(\boldsymbol{s}_{i(l)}, t_l)$ with $l = 1 \dots T$, $i(l) = 1, \dots N_l$.

Let $f_Y(y_{il}, y_{jk})$ the Gaussian density of the bivariate random vector $Y(\boldsymbol{s}_{i(l)}, t_l), Y(\boldsymbol{s}_{j(k)}, t_k)$. Then, the pairwise likelihood function is defined as:

$$pl(\boldsymbol{\theta}) = \sum_{i,j,l,k \in D} log(f_Y(y_{il}, y_{jk}))w_{ijlk} \tag{3}$$

where D is a suitable set index and w_{ijlk} are non-negative weights, not depending on $\boldsymbol{\theta}$, specified as:

$$w_{ijlk} = \begin{cases} 1 & ||\boldsymbol{s}_{i(l)} - \boldsymbol{s}_{j(k)}|| < d_s, |t_l - t_k| < d_t \\ 0 & \text{otherwise} \end{cases} . \tag{4}$$

and in this case $\boldsymbol{\theta} = (\mu, \sigma^2, \alpha_s, \alpha_t)^T$. The pairwise likelihood estimator $\hat{\boldsymbol{\theta}}_{pl}$ is obtained maximizing (3) with respect to $\boldsymbol{\theta}$. In the GeoModels package we can choose the fixed parameters and the parameters that must be estimated. Pairwise likelihood estimation is performed with the function GeoFit:

```
## estimation with pairwise likelihood
fixed=list(nugget=nugget)
start=list(mean=mean,mean1=mean1, sill=sill,
            scale_s=scale_s,scale_t=scale_t)
fit = GeoFit(data=ss1,coordx_dyn=coordx_dyn, coordt=coordt,
  corrmodel=corrmodel,maxdist=0.1,maxtime=1,X=X,
  optimizer="BFGS", start=start,fixed=fixed,model=model)
```

The object fit include informations about the pairwise likelihood estimation:

```
fit
##################################################################
Maximum Composite-Likelihood Fitting of Gaussian Random Fields
Setting: Marginal Composite-Likelihood
Model associated to the likelihood objects: Gaussian
Type of the likelihood objects: Pairwise
Covariance model: Exp_Exp
Number of spatial coordinates: 3266
Number of dependent temporal realisations: 16
Type of the random field: univariate
Number of estimated parameters: 5
Type of convergence: Successful
Maximum log-Composite-Likelihood value: -75462.58
Estimated parameters:
   mean    mean1  scale_s  scale_t      sill
 0.4072  -0.3557   0.1012   1.6485    0.9933
##################################################################
```

Note that the option maxdist=0.1 and maxtime=1 set the compact supports of the weight function (4) i.e. $d_s = 0.1$ and $d_t = 1$.

Checking model assumptions

Given the estimation of the mean regression and sill parameters $\hat{\boldsymbol{\beta}}, \hat{\sigma}^2$, the estimated residuals

$$\widehat{Z}(\boldsymbol{s}_{i(l)}, t_l) = \frac{Y(\boldsymbol{s}_{i(l)}, t_l) - X(\boldsymbol{s}_{i(l)}, t_l)^T \widehat{\boldsymbol{\beta}}}{(\widehat{\sigma}^2)^{\frac{1}{2}}}, \quad l = 1 \ldots T, \quad i = 1, \ldots N_l$$

can be viewed as a realization of zero mean a stationary Gaussian RF with correlation function $\rho(\boldsymbol{h}, u)$. The residuals can be computed using the `GeoResiduals` function:

```
res=GeoResiduals(fit) # computing residuals
```

Then the marginal distribution assumption on the residuals can be graphically checked with a Gauussian qq-plot (Figure 2, left part) using the function `GeoQQ`.

```
### checking model assumptions: marginal distribution
GeoQQ(res)
```

The correlation model assumption can be checked comparing the empirical and the estimated space-time semivariogram functions using the `GeoVariogram` and `GeoCovariogram` functions (Figure 2, right part):

```
### checking model assumptions: ST variogram model
vario = GeoVariogram(data=res$data,coordx_dyn=coordx_dyn,
      coordt=coordt, maxdist=0.6,maxtime=7)
GeoCovariogram(res,vario=vario,fix.lagt=1,fix.lags=1,
            show.vario=TRUE,pch=20)
```

We remark that the space-time empirical semivariogram computation has to be slightly modified with respect to the classical version because the marginal temporal variogram is not defined under our setting.

Prediction of space-time Gaussian random fields

For a given space time location $(\boldsymbol{s}_0, t_0)$ with associated covariates $X(\boldsymbol{s}_0, t_0)$, the optimal prediction of Gaussian RF is computed as:

$$\widehat{Y}(\boldsymbol{s}_0, t_0) = X(\boldsymbol{s}_0, t_0)^T \widehat{\boldsymbol{\beta}} + \sum_{l=1}^{T} \sum_{i=1}^{N_l} \lambda_{l,i} [Y(\boldsymbol{s}_{i(l)}, t_l) - X(\boldsymbol{s}_{i(l)}, t_l)^T \widehat{\boldsymbol{\beta}}] \tag{5}$$

where the vector of weights $\boldsymbol{\lambda} = (\lambda_{1,1}, \ldots, \lambda_{T,N_T})'$ is given by $\boldsymbol{\lambda} = R^{-1}\boldsymbol{c}$ and

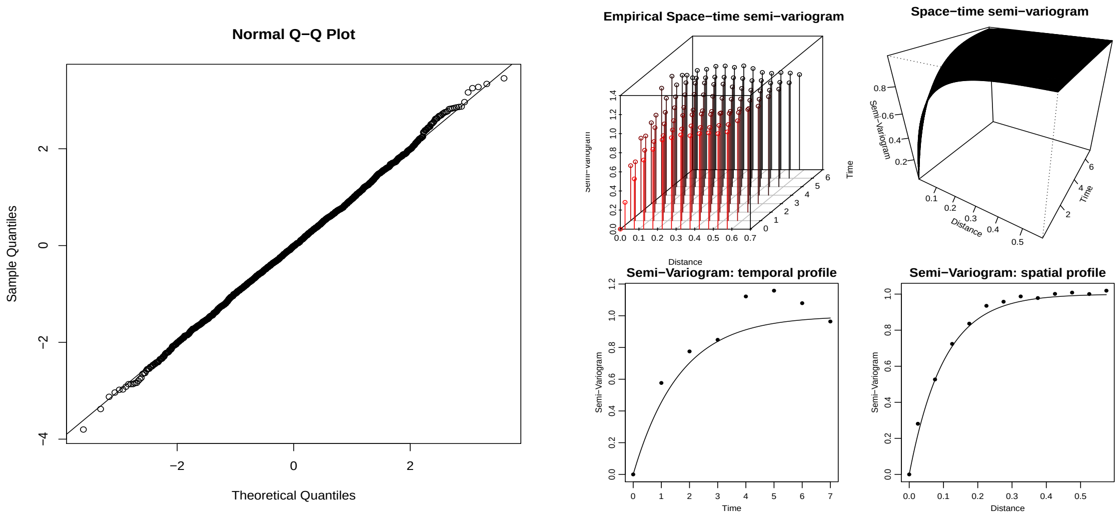

Figure 2: Left: QQ-plot for the residuals of the space-time Gaussian RF. Right: space-time empirical vs estimated semi-variogram function for the residuals

- $\boldsymbol{c} = (cor(Y(\boldsymbol{s}_0, t_0), Y(\boldsymbol{s}_{1(1)}, t_1)), \ldots, cor(Y(\boldsymbol{s}_0, t_0), Y(\boldsymbol{s}_{N_T(T)}, t_T)))^T$.
- $R = [[\text{cor}(Y(\boldsymbol{s}_{i(l)}, t_l), Y(\boldsymbol{s}_{j(k)}, t_k)]_{l,k=1}^T]_{i,j=1}^{N_l, N_k}$ is the correlation matrix.

Kriging can be performed using the `GeoKrig` function. We need just to specify the spatial location and temporal instants to predict. In this example we consider a spatial regular grid and the first two temporal instants:

```
 ## spatial locations to predict
xx=seq(0,1,0.02)
loc_to_pred=as.matrix(expand.grid(xx,xx))
times=c(coordt[1],coordt[2])
```

Additionally, we need to specify the associated covariates:

```
Nloc=nrow(loc_to_pred)*length(times)
Xloc=cbind(rep(1,Nloc),runif(Nloc))
```

Then the optimal linear prediction (5), using the estimated parameters, can be performed using the `GeoKrig` function:

```
param_est=as.list(c(fit$param,fixed))
pr = GeoKrig(data=ss1,coordx_dyn=coordx_dyn, coordt=coordt,
     corrmodel=corrmodel,X=X,Xloc=Xloc,model=model,
     loc=loc_to_pred,time=times,param=param_est)
```

A kriging map for the first two temporal instants (Figure 3) with a comparison with the observed data can be obtained with the following code:

```
par(mfrow=c(length(times),2))
colour = rainbow(100)
i=1
for(i in 1:length(times)) {
quilt.plot(coordx_dyn[[i]],ss1[[i]],col=colour)
image.plot(xx, xx, matrix(pr$pred[i,],ncol=length(xx)),col=colour,
            main = paste("Kriging␣Time=" , times[i]),ylab="")
}
```

References

Bevilacqua, M. and V. Morales-Oñate (2018). *GeoModels: A Package for Geostatistical Gaussian and non Gaussian Data Analysis*. R package version 1.0.3-4.

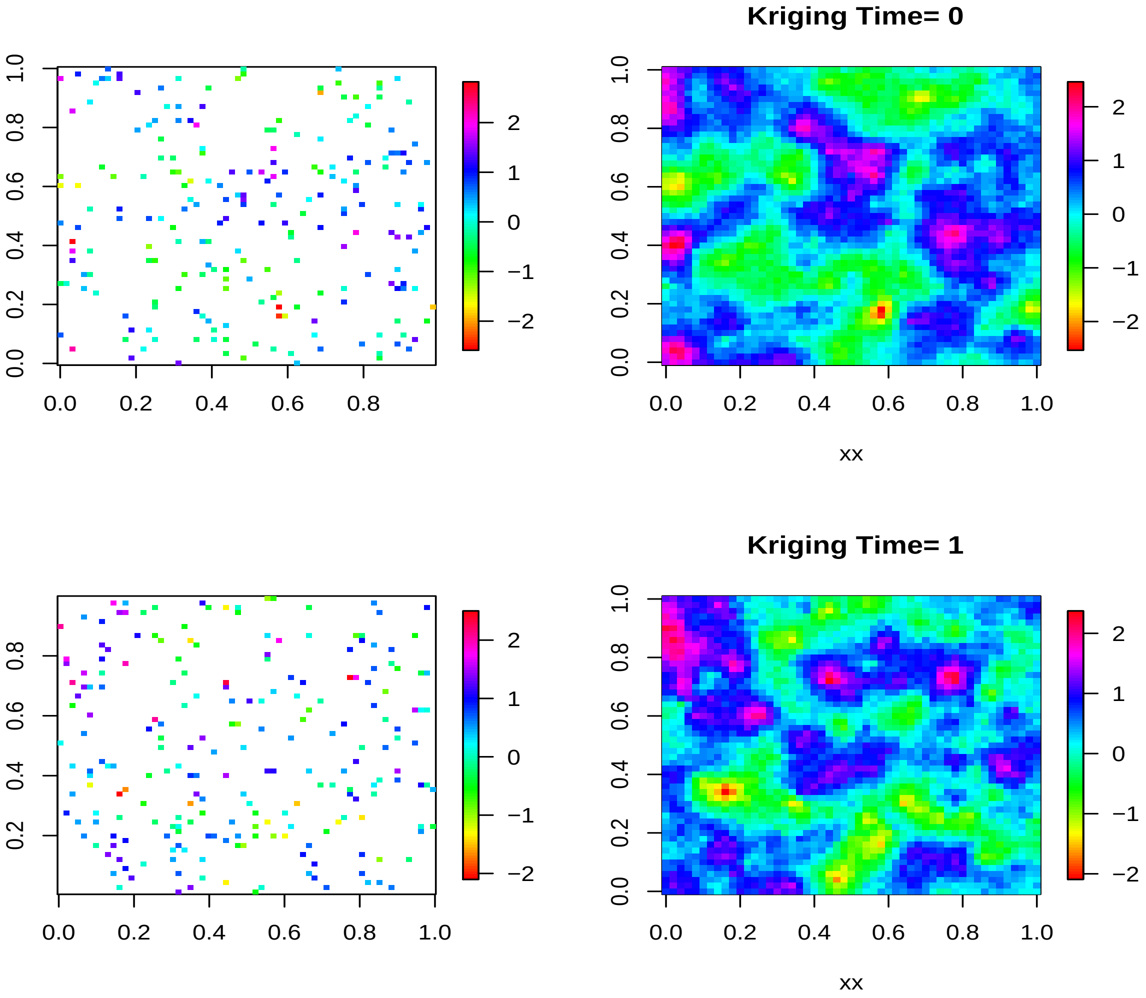

Figure 3: Gaussian space-time kriging for the first two temporal instants