

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

Moreno Bevilacqua  
V ctor Morales-O ate

August 22, 2018

## 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(121)
```

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

Let us consider a space-time Gaussian RF  $Z = \{Z(\mathbf{s}, t), \mathbf{s} \in S, t \in B\}$ , where  $\mathbf{s}$  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  $\rho(\mathbf{h}, u) = \text{cor}(Z(\mathbf{s} + \mathbf{h}, t + u), Z(\mathbf{s}, t))$ .

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

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

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

Suppose we want to simulate a realization of  $Y$  at  $t_1 = 0, t_2 = 0.5, \dots, t_T = 8, T = 17$  temporal instants and  $N_1, \dots, N_T$  spatial locations (changing over time) 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(0,8,0.5) # Define the temporal coordinates
coordx_dyn=list(); X=list()
maxN=180
for(k in 1:length(coordt))
{
NN=sample(1:maxN,size=1)
x <- runif(NN, 0, 1); y <- runif(NN, 0, 1)
coordx_dyn[[k]]=cbind(x,y) # spatial matrix coordinates for each time
X[[k]]=cbind(rep(1,NN),runif(NN)) # spatial matrix covariates for each time
}
```

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

```
unlist(lapply(coordx_dyn,nrow))
[1] 72 150 55 159 48 75 169 56 89 178 27 168 70 175 23 40 118
```

and the total number of space-time locations is given by  $\sum_{i=1}^T N_i = N$ , in our example  $N = 1672$ .

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= -0.8
sill = 1; nugget = 0
```

where `mean`, `mean1` and `sill` are respectively  $\beta_1$ ,  $\beta_2$  and  $\sigma^2$ .

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

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

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

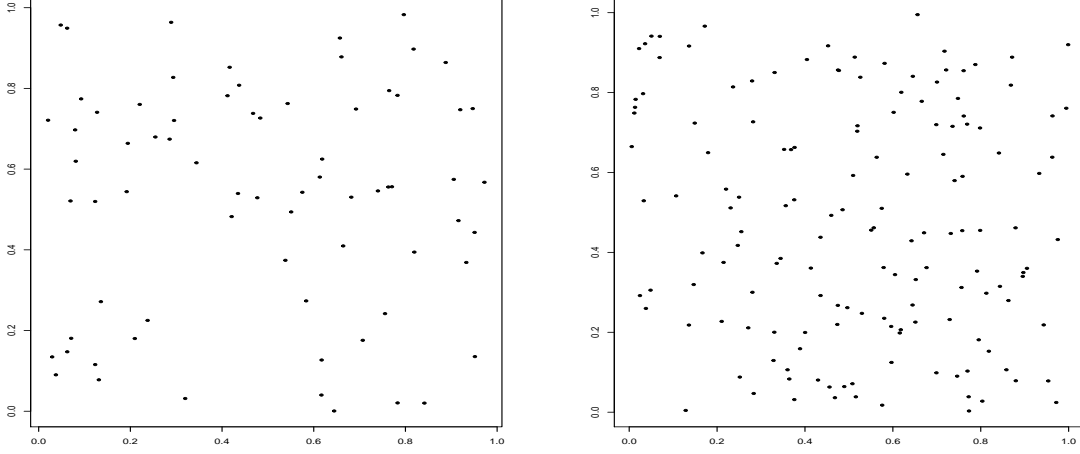


Figure 1: Spatial coordinates for the first two temporal instants

```
corrmodel = "Exp_Exp"
scale_s = 0.2/3
scale_t = 1/3
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. If the spatial coordinates are fixed over time then we need to set the option `coordx` as a  $N \times 2$  matrix.

## Estimation of Gaussian space-time random fields

Given a space-time realization  $\{Y(\mathbf{s}_i, t_l), \quad l = 1 \dots T, i = 1, \dots N_l\}$ , let  $f_Y(y_{il}, y_{jk})$  the Gaussian density of the bivariate random vector  $Y(\mathbf{s}_i, t_l), Y(\mathbf{s}_j, t_k)$ . Then, the pairwise likelihood function is defined as:

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

where

$$D = \begin{cases} l = 1 \dots T, & i = 1, \dots, N_l, & k = l, \dots, T \\ j = i + 1, \dots, N_l & \text{if } l = k \\ j = 1, \dots, N_k & \text{if } l > k \end{cases}.$$

and  $w_{ijkl}$  are non-negative weights, not depending on  $\theta$ , specified as:

$$w_{ijkl} = \begin{cases} 1 & ||s_i - s_j|| < d_s, |t_l - t_k| < d_t \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

and in this case  $\theta = (\mu, \sigma^2, \alpha_s, \alpha_t)^T$ . The pairwise likelihood estimator  $\hat{\theta}_{pl}$  is obtained maximizing (3) with respect to  $\theta$ . 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, 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: 1672
Number of dependent temporal realisations: 17
Type of the random field: univariate
Number of estimated parameters: 5
Type of convergence: Successful
Maximum log-Composite-Likelihood value: -35878.36
```

Estimated parameters:

mean	mean1	scale_s	scale_t	sill
0.19602	-0.79176	0.07457	0.34477	1.08394

#####

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  $\hat{\beta}, \hat{\sigma}^2$ , the estimated residuals

$$\widehat{Z(s_i, t_l)} = \frac{Y(s_i, t_l) - X(s_i, t_l)^T \hat{\beta}}{(\hat{\sigma}^2)^{\frac{1}{2}}} \quad l = 1 \dots T, i = 1, \dots N_l$$

can be viewed as a realization of zero mean a stationary Gaussian RF with correlation function  $\rho(\mathbf{h}, u)$ . 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 for instance with a qq-plot (Figure 2, left part):

```
### checking model assumptions: marginal distribution
qqnorm(unlist(res$data))
abline(0,1)
```

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=4)
GeoCovariogram(res, vario=vario, fix.lagt=1, fix.lags=1, show.vario=TRUE, pch=20)
```

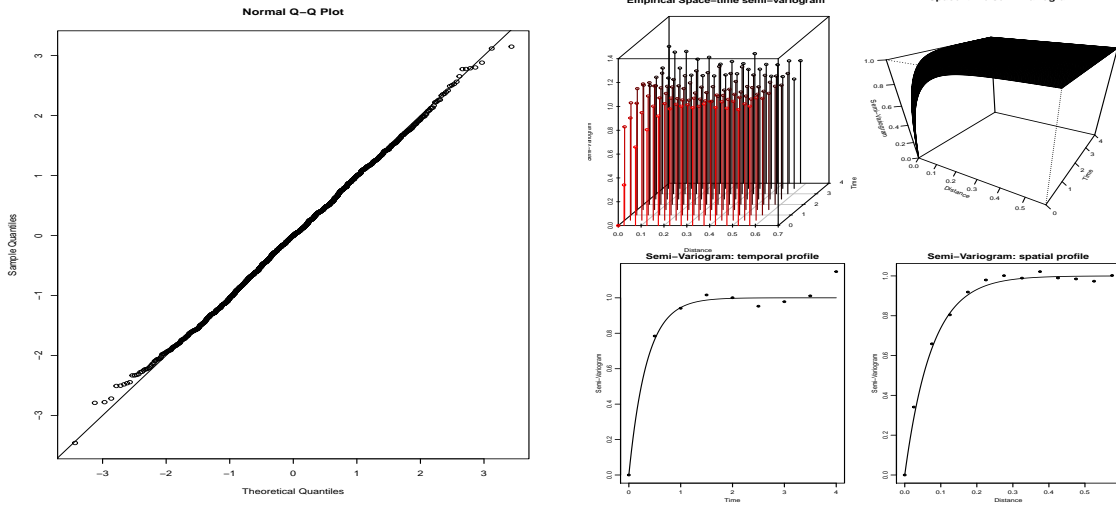


Figure 2: Left: QQ-plot for the residuals of the space-time Gaussian RF. Right: space-time empirical vs estimated semi-variogram function for the residuals

## Prediction of space-time Gaussian random fields

For a given space time location  $(s_0, t_0)$  with associated covariates  $X(s_0, t_0)$ , the optimal prediction of Gaussian RF is computed as:

$$\hat{Y}(s_0, t_0) = X(s_0, t_0)^T \hat{\beta} + \sum_{l=1}^T \sum_{i=1}^{N_l} \lambda_{l,i} [Y(s_i, t_l) - X(s_i, t_l)^T \hat{\beta}] \quad (5)$$

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

- $c = (cor(Y(s_0, t_0), Y(s_1, t_1)), \dots, cor(Y(s_0, t_0), Y(s_{N_T}, t_T)))^T$ .
- $R = [[cor(Y(s_i, t_l), Y(s_j, t_k))]_{l,k=1}^T]_{i,j=1}^{N_l, N_k}$  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 two temporal instants:

```
## spatial locations to predict
xx=seq(0,1,0.03)
loc_to_pred=as.matrix(expand.grid(xx,xx))
## temporal instants to predict
times=c(0.5,1.5)
```

Moreover we need to specify the associated covariates as a list

```

Nloc=nrow(loc_to_pred)
Xloc=list()
Xloc[[1]]=cbind(rep(1,Nloc),runif(Nloc)) # covariates for the first time
Xloc[[2]]=cbind(rep(1,Nloc),runif(Nloc)) # covariates for the second time

```

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,
              corrmmodel=corrmmodel, X=X,Xloc=Xloc, model=model,
              mse=TRUE, loc=loc_to_pred,time=times,param=param_est)

```

A kriging map for the two temporal instants with associate mean square error (Figure 3) can be obtained with the following code:

```

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



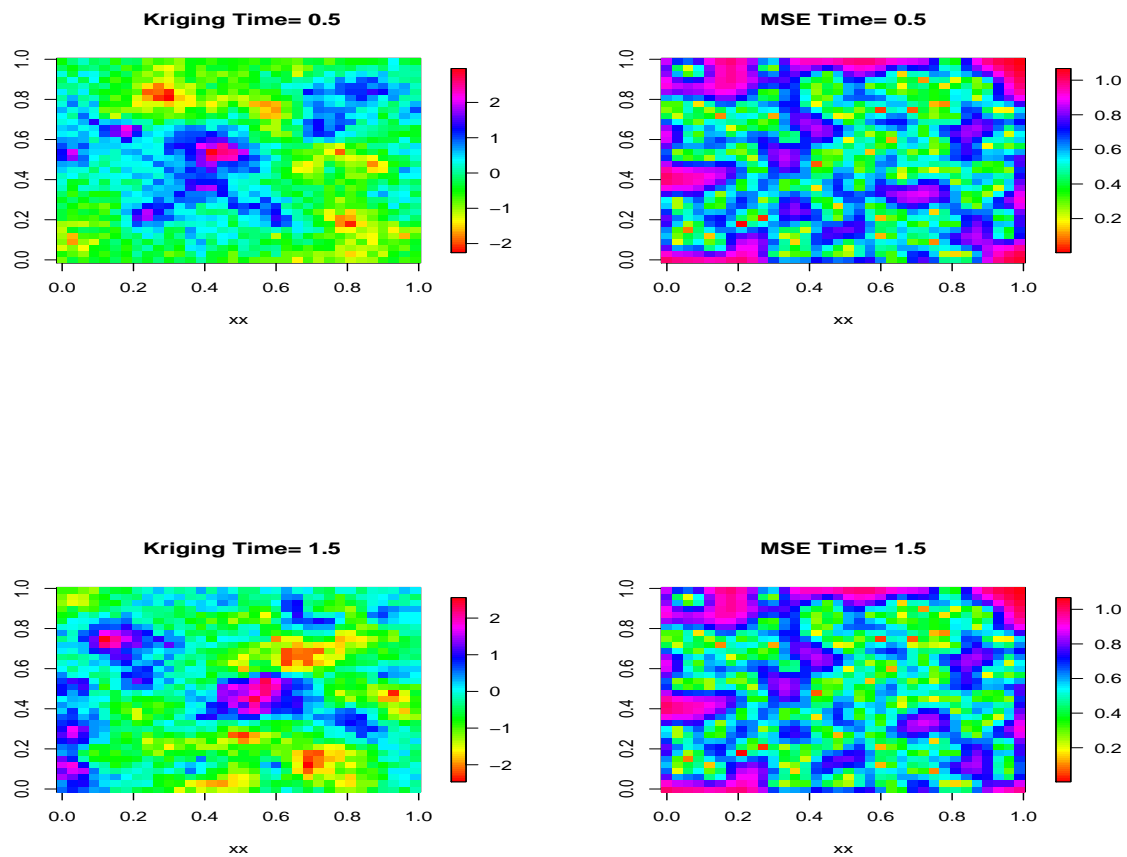


Figure 3: Gaussian space-time kriging for two temporal instants and associated mean square error