

GeoModels Tutorial: simulation, estimation and prediction of spatio-temporal data using Gaussian random fields

Moreno Bevilacqua
V ctor Morales-O ate

Introduction

In this tutorial we show how to analyze geo-referenced spatio temporal data using Gaussian random fields (RFs) with the R package `GeoModels`.

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-OCL")
require(GeoModels)
require(fields)
model="Gaussian" # model name in the GeoModels package
set.seed(12)
```

Simulation of a space-time Gaussian random field

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 $t_1 = 1, t_2 = 2, \dots, t_T = 30$, $T = 30$ temporal instants and $N = 400$ spatial locations uniformly distributed in the unit square. The total number of space-time locations is given by $NT = 12000$.

We first set the temporal instants and then the spatial coordinates with associated covariates.

```
coordt=1:30 # number of temporal instants
```

```

T=length(coordt)
NN=400          # number of spatial locations
x = runif(NN, 0, 1); y = runif(NN, 0, 1)
coords=cbind(x,y)
X=cbind(rep(1,NN*T),runif(NN*T))

```

We then specify the mean, variance and nugget parameters

```

mean = 0.5; mean1= -0.25
sill=2; nugget=0

```

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

In this tutorial we assume a simple spatially isotropic and symmetric in time separable Wendland model for the space-time correlation function that is:

$$\rho((\mathbf{h}, u); \alpha_s, \alpha_t, \mu_s, \mu_t) = \left(1 - \frac{\|\mathbf{h}\|}{\alpha_s}\right)_+^{\mu_s} \left(1 - \frac{|u|}{\alpha_t}\right)_+^{\mu_t} \quad (2)$$

This kind of correlation model is compactly supported in space or time that is the correlation function is zero when $\|\mathbf{h}\| > \alpha_s$ or $|u| > \alpha_t$. This is an interesting feature since algorithms for sparse matrices can be used to speed-up the computations associated with the corresponding correlation matrix.

Then we set the name of the correlation model and the associated parameters:

```

corrmodel="Wend0_Wend0";
scale_s=0.2; scale_t=2

```

where `scale_s` and `scale_t` corresponds to α_s and α_t , the compact supports of the correlation model. We are now ready to simulate, through Cholesky decomposition, the space time Gaussian RF using the function `GeoSim`:

```

param= list(nugget=0,mean=mean,mean1=mean1, scale_t=scale_t,
            scale_s=scale_s,sill=sill,power2_s=4,power2_t=4)
ss1 = GeoSim(coordx=coords, coordt=coordt, corrmodel=corrmodel,
            X=X,sparse=TRUE, model=model,param=param)$data

```

Note that the option `sparse=TRUE` allows to consider algorithms for sparse matrices when performing Cholesky decomposition, using in the package `spam` (Furrer and Sain (2010)). Informations about the sparsity of the covariance matrix can be obtained through the function `GeoCovmatrix` with the following code

```
cc = GeoCovmatrix(coordx=coords, coordt=coordt, corrmodel=corrmodel,
                  X=X, sparse=TRUE, model=model, param=param)
[1] TRUE
cc$nozero
[1] 0.01768
```

This means that (approximatively) 98% of the covariance matrix are zeros *i.e* the matrix is highly sparsed.

Estimation of a Gaussian space-time random field

Given a space-time realization $\{Y(\mathbf{s}_i, t_l), \quad l = 1 \dots T, \quad i = 1, \dots, N\}$, let $f_Y(y_{il}, y_{jk})$ the Gaussian density of a pair of observations $Y(\mathbf{s}_i, t_l)$ and $Y(\mathbf{s}_j, t_k)$. Then, the pairwise likelihood function is defined as (Bevilacqua and Gaetan, 2015):

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

where

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

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

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

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` using the BFGS method:

```
start=list(mean=mean, mean1=mean1, scale_s=scale_s,
           scale_t=scale_t, sill=sill)
fixed=list(nugget=nugget, power2_s=4, power2_t=4)
fit = GeoFit(data=ss1, coordx=coords, coordt=coordt, corrmodel=corrmodel,
```

```

maxdist=0.05,maxtime=1,X=X,
optimizer="BFGS",start=start,fixed=fixed)

```

Note that the option `maxdist=0.05` and `maxtime=1` set the (arbitrary) compact supports of the weight function (4) i.e. $d_s = 0.05$ and $d_t = 1$. A suitable choice of the weights allows to improve both the statistical and computational efficiency (Bevilacqua and Gaetan (2015)).

The object `fit` include informations about the pairwise likelihood estimation

```

fit
#####
Maximum Composite-Likelihood Fitting of Gaussian Random Fields
Setting: Marginal Composite-Likelihood
Model: Gaussian
Type of the likelihood objects: Pairwise
Covariance model: Wend0_Wend0
Optimizer: BFGS
Number of spatial coordinates: 400
Number of dependent temporal realisations: 40
Type of the random field: univariate
Number of estimated parameters: 5
Type of convergence: Successful
Maximum log-Composite-Likelihood value: -237670.62
Estimated parameters:
      mean    mean1  scale_s  scale_t      sill
0.5072  -0.2624   0.1981   2.0743   1.9629
#####

```

Checking model assumptions

Given the estimation of the mean regression and sill parameters, the estimated residuals

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

can be viewed as a realization of a zero mean 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 1, 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 1, right part):

```
### checking model assumptions: ST semi-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)
```

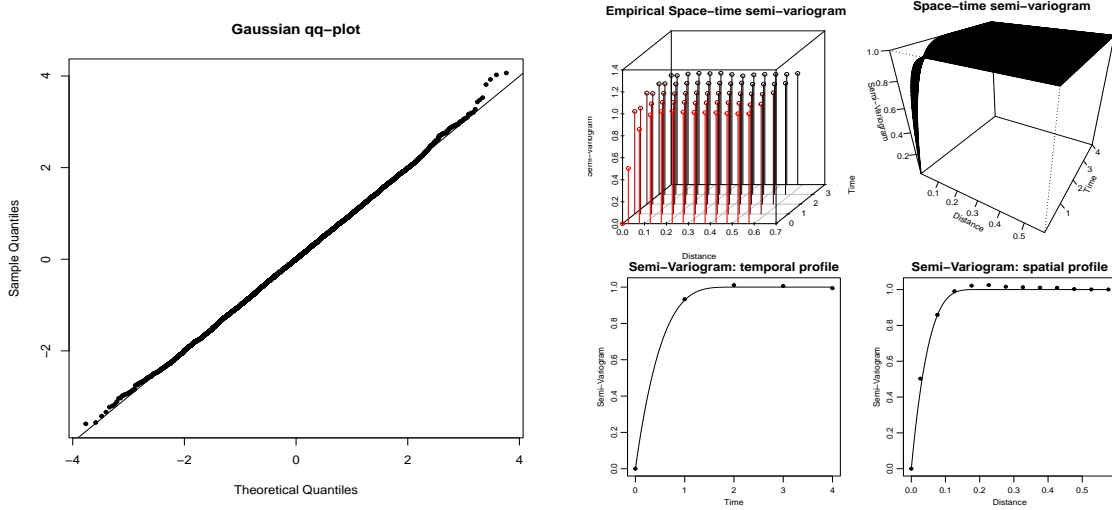


Figure 1: Left: QQ-plot for the residuals of the space-time Gaussian RF. Right: space-time empirical vs estimated semi-variogram function for the residuals

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:

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

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

- $\mathbf{c} = (\text{cor}(Y(\mathbf{s}_0, t_0), Y(\mathbf{s}_1, t_1)), \dots, \text{cor}(Y(\mathbf{s}_0, t_0), Y(\mathbf{s}_N, t_T)))^T$.
- $R = [[\text{cor}(Y(\mathbf{s}_i, t_l), Y(\mathbf{s}_j, t_k))]_{l,k=1}^T]_{i,j=1}^N$ is the correlation matrix.

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

```
xx=seq(0,1,0.03)
loc_to_pred=as.matrix(expand.grid(xx,xx)) # locations to predict
n_loc=nrow(loc_to_pred)
times=c(2) #time to predict
```

In addition, we need to specify the associated covariates:

```
Xloc=cbind(rep(1,n_loc),runif(n_loc))
```

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=coords, coordt=coordt, corrmmodel,
             X=X,Xloc=Xloc,sparse=TRUE,
             model=model,mse=TRUE,loc=loc_to_pred,
             time=times,param=param_est)
```

A kriging map for the temporal instant $t = 2$ with associate mean square error (Figure 2) can be obtained with the following code:

```
par(mfrow=c(1,3))
colour <- rainbow(100)
## observed data at time 2
quilt.plot(coords[,1],coords[,2],ss1[2,],col=colour,main="Time=2")
## kriging map at time 2
image.plot(xx, xx, matrix(pr$pred,ncol=length(xx)),col=colour,
           main = paste("Kriging-Time=" , 2),ylab="")
## MSE kriging map at time 2
image.plot(xx, xx, matrix(pr$mse,ncol=length(xx)),col=colour,
           main = paste("Stderr-Time=" , 2),ylab="")
```

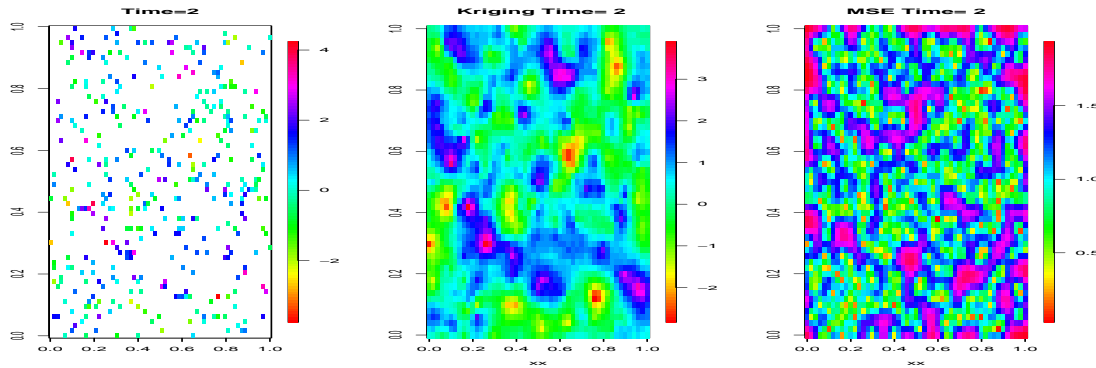


Figure 2: From left to right: observed spatial data at time $t = 2$, associated kriging map and mean square error map.

References

- Bevilacqua, M. and C. Gaetan (2015). Comparing composite likelihood methods based on pairs for spatial Gaussian random fields. *Statistics and Computing* 25, 877–892.
- Furrer, R. and S. R. Sain (2010). spam: a sparse matrix R package with emphasis on mcmc methods for Gaussian Markov random fields. *Journal of Statistical Software* 36, 1–25.