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

Moreno Bevilacqua
Víctor Morales-Oñate

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(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 $Z = \{Z(\boldsymbol{s}, t), \boldsymbol{s} \in S, t \in B\}$, 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 $t_1 = 1, t_2 = 2, \ldots, t_T = 30$, $T = 30$ 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:

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

This kind of correlation model is compactly supported in space or time that is the correlation function is zero when $||\boldsymbol{h}|| > \alpha_s$ 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 decompositon, 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 though 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 $\{Y(\boldsymbol{s}_i, t_l), \quad l = 1 \dots T,\ i = 1, \dots N\}$, let $f_Y(y_{il}, y_{jk})$ the Gaussian density of a pair of observations $Y(\boldsymbol{s}_i, t_l)$ and $Y(\boldsymbol{s}_j, t_k)$. Then, the pairwise likelihood function is defined as (Bevilacqua and Gaetan, 2015):

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

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_{ijlk} are non-negative weights, not depending on $\boldsymbol{\theta}$, specified as:

$$w_{ijlk}(m, d_t) = \begin{cases} 1 & \mathbf{s}_i \in N_m(\mathbf{s}_j), |t_l - t_k| < d_t \\ 0 & \text{otherwise} \end{cases}. \tag{4}$$

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` 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, neighb =5,maxtime=1,X=X,
             optimizer="BFGS",start=start,fixed=fixed)
```

Note that the option `neighb=5` and `maxtime=1` set the (arbitrary) compact supports of the weight function (4) i.e. $m = 5$ and $d_t = 1$.

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
Distance: Eucl
Type of the likelihood objects: Pairwise
Covariance model: Wend0_Wend0
Optimizer: BFGS
Number of spatial coordinates: 400
Number of dependent temporal realisations: 30
Type of the random field: univariate
Number of estimated parameters: 5
Type of convergence: Successful
Maximum log-Composite-Likelihood value: -610875.85
Estimated parameters:
   mean    mean1  scale_s  scale_t     sill
 0.5232  -0.2841   0.2010   2.0634   1.9699
##################################################################
```

Checking model assumptions

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

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

can be viewed as a realization of a zero mean 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 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)
```

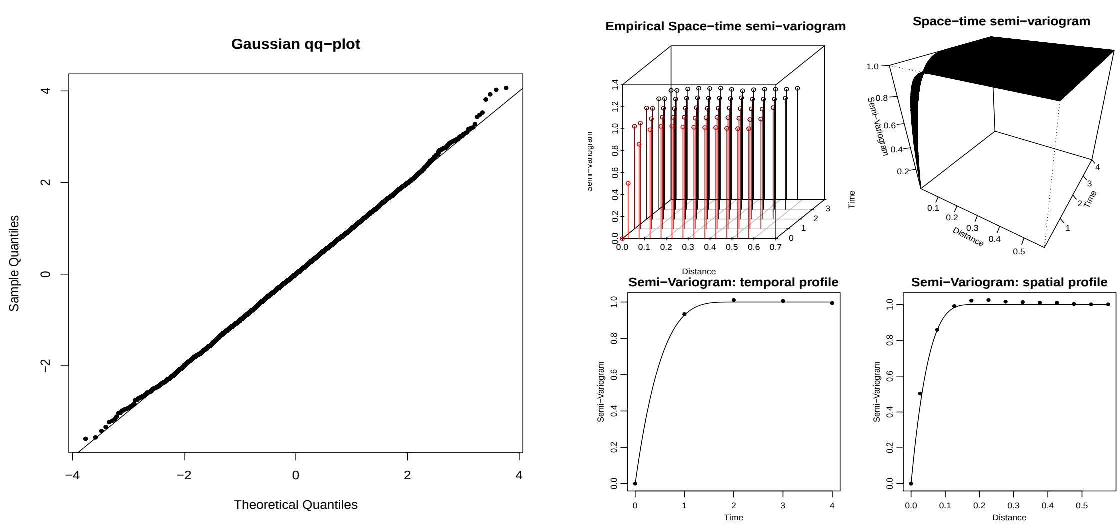

Figure 1: Left: QQ-plot for the residuals of the space-time Gaussian RF. Right: space-time empirical vs estimated semi-variogram function for the residuals

Prediction of space-time Gaussian random fields

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

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

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

- $\boldsymbol{c} = (cor(Y(\boldsymbol{s}_0, t_0), Y(\boldsymbol{s}_1, t_1)), \ldots, cor(Y(\boldsymbol{s}_0, t_0), Y(\boldsymbol{s}_N, t_T)))^T$.

- $R = [[\text{cor}(Y(\boldsymbol{s}_i, t_l), Y(\boldsymbol{s}_j, t_k)]_{l,k=1}^T]]_{i,j=1}^N$ 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:

```
pr = GeoKrig(fit,loc=loc_to_pred,time=times,Xloc=Xloc,sparse=TRUE,mse=TRUE)
```

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="")
```

Local spacetime kriging can be an alternative in order to speedup the computation of the kriging predictions using a fixed space time neighboord. It can be performed using the `GeoKrigloc` function that can be parallelized:

```
pr1 = GeoKrigloc(fit,loc=loc_to_pred,time=times,Xloc=Xloc,sparse=TRUE,
mse=TRUE,neighb=150,maxtime=2,parallel=TRUE)
```

It can be appreciated that classical and local kriging predictions, in this example, are pretty similar

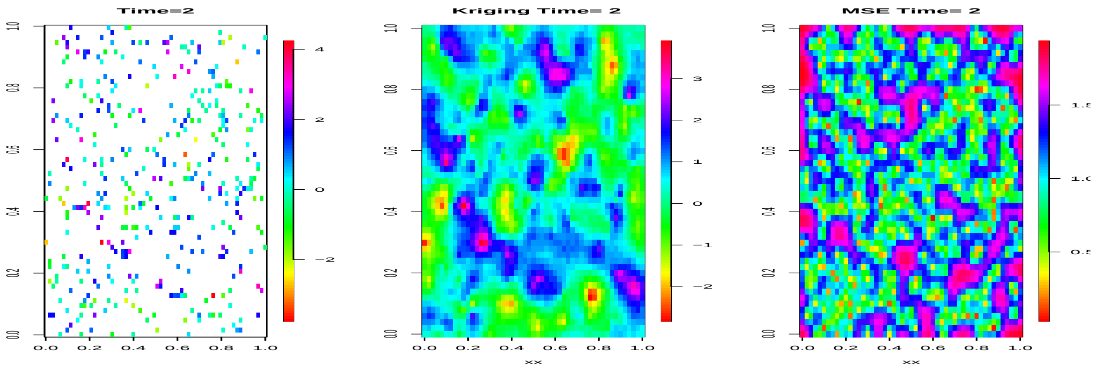

Figure 2: From left to right: observed spatial data at time $t = 2$, associated kriging map and mean square error map.

```
summary(c(pr$pred-prloc$pred))
  Min.     1st Qu.      Median         Mean     3rd Qu.         Max.
-1.167e-04  -5.526e-07   6.700e-10   9.572e-07   1.157e-06   2.031e-04
```

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 rields. *Journal of Statistical Software 36*, 1–25.