`GeoModels` Tutorial: analysis of global spatial data on the planet Earth using Gaussian random fields

Moreno Bevilacqua

Introduction

This tutorial illustrates how to analyze spherical data with the `R` package `GeoModels` (Bevilacqua and Morales-Oñate (2018)). Albeit the main focus of the tutorial is on Gaussian random fields (GRFs) defined over spheres, any of the non GRF shown in the package `GeoModels` can be used for the analysis of data collected over spheres.

We denote with $\mathbb{S}^2$ the unit sphere of $\mathbb{R}^3$, defined as

$$\mathbb{S}^2 = \{\boldsymbol{x} \in \mathbb{R}^3 : ||\boldsymbol{x}|| = R\}.$$

Since a sphere of radius $R = 6371$ is a satisfactory approximation of planet Earth, the tutorial focuses on this specific sphere and shows how to simulate, estimate and predict GRFs on planet Earth.

We start by loading the libraries needed for the analysis and set the radius of the sphere:

```
rm(list=ls())
library(GeoModels)
library(mapproj)
library(globe)
library(fields)
library(sphereplot)
radius=6371
set.seed(1891)
```

Gaussian random fields defined on planet Earth

We consider a zero mean and unit variance GRF $Y = \{Y(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{S}^2\}$ defined on a sphere of radius $R = 6371$. A point on $\mathbb{S}^2$ can be parametrized trough spherical coordinates using radians as (R, lon^*, lat^*), where $lon^* \in [-\pi, \pi]$ is the longitude and $lat^* \in [-\pi/2, \pi/2]$ is the latitude. Alternatively, one can use decimal degree (R, lon, lat) where $lon \in [-180, 180]$ and $lat \in [-90, 90]$ (the `GeoModels` package uses the second parameterization).

The great circle (GC) distance corresponds to the shortest path joining two points on the spherical surface and it is the natural distance to be used when analyzing data on a sphere. Alternatively, one can compute the chordal (CH) distance, that is the segment below the arc joining any pair of points located over the spherical shell. Let us give some details on the computation of GC and CH distances. Given two location sites in longitude

and latitude (expressed in decimal degrees), that is, $\boldsymbol{x}_i = (\text{lon}_i, \text{lat}_i)$ and $\boldsymbol{x}_j = (\text{lon}_j, \text{lat}_j)$, and given the radius R of the Earth, the GC distance, d_{GC}, is given by:

$$d_{GC}(\boldsymbol{x}_i, \boldsymbol{x}_j) = R\theta_{ij},$$

where

$$\theta_{ij} = [\arccos\{\sin a_i \sin a_j + \cos a_i \cos a_j \cos(b_i - b_j)\}]$$

is the GC distance on the unit sphere. Here, $a_i = (\text{lat}_i)\pi/180$, $a_j = (\text{lat}_j)\pi/180$, $b_i = (\text{lon}_i)\pi/180$, $b_j = (\text{lon}_j)\pi/180$ and $\mathbf{x}_i = (a_i, b_i)^T$ and $\mathbf{x}_j = (a_j, b_j)^T$.

The CH distance is instead uniquely defined through the relation

$$d_{CH}(\boldsymbol{x}_i, \boldsymbol{x}_j) = 2Rsin(\theta_{ij}/2). \tag{1}$$

CH obviously underestimates the GC distance, and the approximation error increases with the size of the considered portion of the planet.

We assume $\text{cor}(Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_j))) = \rho(d_{GC}(\boldsymbol{x}_i, \boldsymbol{x}_j))$ where $\rho : [0, R\pi] \to \mathbb{R}$ is continuous with $\rho(0) = 1$. Under this setting, the correlation function is called *geodesically isotropic* (Gneiting, 2013; Porcu et al., 2016). We then consider the location and scale transformation

$$Z(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \sigma Y(\boldsymbol{x})$$

where $\mu(\boldsymbol{x}) \in \mathbb{R}$ is a spherical varying mean and $\sigma > 0$. The package `GeoModels` allows a mean regression specification, that is $\mu(\boldsymbol{x}) = M(\boldsymbol{x})^T\boldsymbol{\beta}$ where $M(\boldsymbol{x})$ is a vector of spherical covariates. Nevertheless, for the sake of simplicity this tutorial assumes a constant mean: $\mu(\boldsymbol{x}) = \mu$.

Then $\mathbb{E}(Z(\boldsymbol{x})) = \mu$ and $\text{cov}(Z(\boldsymbol{x}_i), Z(\boldsymbol{x}_j))) = \sigma^2\rho(d_{GC}(\boldsymbol{x}_i, \boldsymbol{x}_j))$ where σ^2 is the variance parameter and in order to obtain a realization from a spherical RF we need to specify a mean, a variance parameter and a valid (*i.e.*positive definite) geodesically isotropic parametric correlation function $\rho_\gamma(\cdot)$.

We first set the spherical coordinates in decimal degrees using the function `pointsphere` of the package `sphereplot` (Robotham, 2013):

```
NN=1500 ## number of location sites on the planet Earth
coords=pointsphere(NN,c(-180,180),c(-90,90),c(radius,radius))[,1:2]
```

The locations on planet Earth can be visualized (Figure 1) using some functions of the package `globe` (Baddeley et al., 2017) with the following code:

```
globeearth(eye=place("newyorkcity"))
globepoints(loc=coords,pch=20,cex=0.4)
globeearth(eye=place("everest"))
globepoints(loc=coords,pch=20,cex=0.4)
```

Figure 1: Location sites on the planet Earth used in this tutorial

As for the correlation function, we assume an Askey model. This is a valid correlation model in $\mathbb{S}^2$ (Gneiting, 2013):

$$\rho_{\alpha,\delta}(d_{GC}) = \begin{cases} (1 - d_{GC}/\alpha)^{\delta} & \theta < \alpha \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

This model is compactly supported provided $\alpha < R\pi$ which means that the associated covariance matrix is sparse, an interesting feature from computational point of view. We set the Askey model with associated parameters and the other parameters of the GRF.

```
corrmodel = "Wend0"    ## correlation model and parameters
scale=radius*0.4
power2=4
sill=1
nugget=0
mean=0
```

where the `scale` parameter corresponds to α (the compact support of the correlation model) and `sill` is the variance parameter σ^2. We are now ready to simulate a GRF on the sphere using the function `GeoSim`:

```
param=list(mean=mean, sill=sill, nugget=nugget,
                scale=scale, power2=power2)
data = GeoSim(coordx=coords, corrmodel=corrmodel, param=param,
                   distance="Geod",sparse=TRUE,radius=radius)$data
```

Note that the option `distance="Geod"` coupled with the location sites given in lon/lat format in degimal degree (`coordx=coords`) and a valid correlation model on the sphere (`corrmodel`) with a given `radius`, allows to perform a simulation based on Cholesky decomposition of the GRF using GC.

Estimation of Gaussian random fields on planet Earth

Given a realization $\boldsymbol{Z} = (Z(\boldsymbol{x}_1), Z(\boldsymbol{x}_2), \ldots, Z(\boldsymbol{x}_N))^T$ from a GRF defined on $\mathbb{S}^2$ with correlation (2), we can perform maximum likelihood estimation maximizing the log-likelihood function

$$l(\boldsymbol{\beta}) = -0.5log(|\sigma^2 C_{\alpha,\delta}|) - 0.5\frac{(\boldsymbol{Z} - \mu\boldsymbol{1})^T C_{\alpha,\delta}^{-1}(\boldsymbol{Z} - \mu\boldsymbol{1})}{\sigma^2}$$

with respect to $\boldsymbol{\beta} = (\mu, \sigma^2, \alpha, \delta)^T$ and where $C_{\alpha,\delta} = [\rho_{\alpha,\delta}(d_{GC}(\boldsymbol{x}_i, \boldsymbol{x}_j))]_{i,j=1}^N$ is the correlation matrix associated to (2).

We can use the `GeoFit` function to perform maximum likelihood estimation (we choose the fixed parameters and the parameters that must be estimated)

```
fixed<-list(nugget=nugget,power2=power2)
start<-list(mean=mean,scale=scale,sill=sill)
fit_geo_ml <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
              likelihood="Full",type="Standard",sparse=TRUE,
              start=start,fixed=fixed,distance="Geod",radius=radius)
```

Note that the option `sparse=TRUE` allows to exploit specific algorithms for sparse matrices implemented in the `spam` package (Furrer and Sain (2010)) when performing maximum likelihood estimation.

Computation of the GC distances involved in the covariance matrix (given in kms) can be obtained using the function `rdist.earth` of the package `fields` (Douglas Nychka et al., 2015):

```
geod_ds1=rdist.earth(coords,miles=F,R=1)
geod_ds=geod_ds1*radius
max_geod=max(geod_ds)
```

Alternatively, pairwise likelihood estimation is obtained by maximizing the function

$$pl(\boldsymbol{\beta}) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} log(f_z(z_i, z_j; \boldsymbol{\beta}))w_{ij} \tag{3}$$

with respect to $\boldsymbol{\beta}$. It can be performed with the following code

```
fit_geo <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
              likelihood="Marginal",type="Pairwise",maxdist=max_geod/70,
              start=start,fixed=fixed,distance="Geod",radius=radius)
```

In (3) $f_z(z_i, z_j; \boldsymbol{\beta})$ is the Gaussian bivariate density and c_{ij} are non-negative weights, not depending on $\boldsymbol{\beta}$, specified as:

$$w_{ij} := \begin{cases} 1 & d_{GC}(\boldsymbol{x}_i, \boldsymbol{x}_j) < d^* \\ 0 & \text{otherwise} \end{cases}. \tag{4}$$

Note that the option `maxdist=max_geod/70` sets the compact support d^* of the weight function (4) as a fraction of the maximum GC distance (285 KM approximatively). A suitable choice of the weights allows to improve both the statistical and computational efficiency oh the pairwise likelihood method (Bevilacqua and Gaetan (2015)).

The two estimates can be obtained from the objects `fit_geo_ml` and `fit_geo` as:

```
 fit_geo_ml$param
  mean          scale           sill
-0.0004784      2440.490    1.007568
 fit_geo$param
  mean          scale           sill
0.0354392       2436.313.    1.0045678
```

In the `GeoModels` package, estimation can be performed using also the CH distance, or the Euclidean distance after a suitable projection of the spherical coordinates.

Computation of CH distances involved in the covariance matrix can be obtained by using (1):

```
chor_ds=2*radius*sin(geod_ds1/2)
max_chor=max(chor_ds)
```

and pairwise likelihood estimation using chordal distances can be performed setting `distance="Chor"`:

```
fit_chor <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
            likelihood="Marginal",type="Pairwise",maxdist=max_chor/70,
                start=start,fixed=fixed,distance="Chor")
```

Finally, we consider pairwise likelihood estimation using a suitable projection of the spherical points and the Euclidean distance. We use the function `mapprojec` of the package `mapproj` (for R by Ray Brownrigg et al., 2017) in order to obtain projected coordinates using sinusoidal projection

```
prj=mapproject(coords[,1], coords[,2], projection="sinusoidal")
coords_prj=cbind(prj$x,prj$y)
```

and we can visualize the projected points (Figure 2) with the following code:

```
sinusoidal.proj = map(database= "world", ylim=c(-90,90),
xlim=c(-180,180),col="grey80", fill=TRUE,
plot=FALSE, projection="sinusoidal")
map(sinusoidal.proj)
points(coords_prj,pch=20,cex=0.4)
```

Other types of projections can be chosen in the package `mapproj` such us, for instance, the mercator or cylindrical projections.

Finally, we can estimate with pairwise likelihood using Euclidean distances (first we need to rescale the projected coordinates with the radius)

```
coords_eucl=radius*coords_prj
eucl_ds=dist(coords_eucl)
max_eucl=max(dist(coords_eucl))
fit_eucl <- GeoFit(data=data,coordx=coords_eucl,corrmodel=corrmodel,
     maxdist=max_eucl/70,likelihood="Marginal",type="Pairwise",
     start=start,fixed=fixed,distance="Eucl",radius=radius)
```

A comparison of the three pairwise estimates with the three type of distances can be obtained as:

```
fit_geo$param
        mean               scale          sill
0.03543916              2436.313        1.004568
fit_chor$param
```

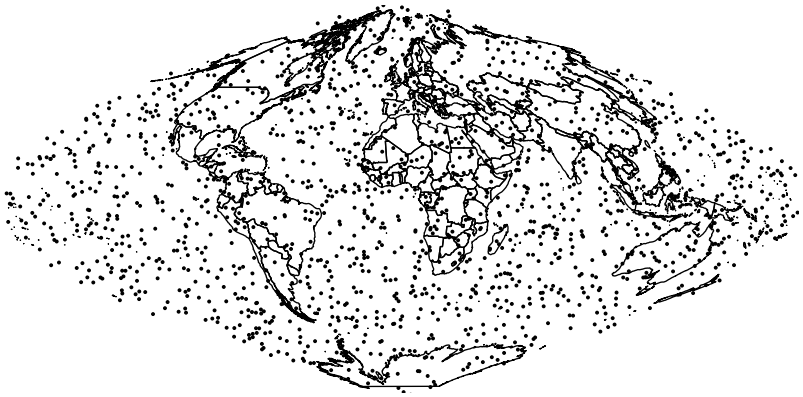

Figure 2: Sinusoidal projection

```
        mean              scale          sill
0.03336823              2328.192       1.038559
fit_eucl$param
        mean              scale          sill
0.04714257              2077.088       0.9466116
```

As expected, the parameter that is most affected by the choice of the distance is the spatial scale parameter. This is not surprising, since CH and euclidean distances are both approximations of the GC distance. Boxplots of the distances (Figure 4) can be useful to show the different behaviour of the three type of distances in this example:

```
boxplot(c(geod_ds),c(chor_ds),c(eucl_ds),
          names=c("Greatcircle","Chordal","Euclidean"))
```

Given the estimation of the mean and variance parameters, the estimated residuals

$$\widehat{Y}(\boldsymbol{x}_i) = \frac{Z(\boldsymbol{x}_i) - \hat{\mu}}{(\widehat{\sigma}^2)^{\frac{1}{2}}} \quad i = 1, \ldots, N$$

can be viewed as a realization of a zero mean unit variance spherical GRF with correlation function (2). Using maximum likelihood estimates, the residuals can be computed using the `GeoResiduals` function:

```
res=GeoResiduals(fit_geo_ml)
```

Then the marginal distribution assumption on the residuals can be graphically checked, for instance, with a qq-plot (Figure 3, left part):

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

The correlation model assumption can be graphically checked comparing the empirical and the estimated semivariogram functions using the `GeoVariogram` and `GeoCovariogram` functions (Figure 3, right part):

```
### checking model assumptions: variogram model
vario = GeoVariogram(data=res$data,coordx=coords,
           maxdist=max_geod/2,distance="Geod")
GeoCovariogram(res,vario=vario,show.vario=TRUE,pch=20)
```

Prediction of Gaussian random fields on planet Earth

For a given location $\boldsymbol{x}_0 \in \mathbb{S}^2$, the optimal prediction in this example is computed by:

$$\widehat{Z}(\boldsymbol{x}_0) = \hat{\mu} + \sum_{i=1}^{N} \lambda_i [Z(\boldsymbol{x}_i) - \hat{\mu}] \tag{5}$$

where $\hat{\mu}$ is a mean estimation and the vector of weights $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)'$ is given by $\boldsymbol{\lambda} = \widehat{C}_{\hat{\alpha},\delta}^{-1} \hat{\boldsymbol{c}}_{\hat{\alpha},\delta}$ where $\widehat{C}_{\hat{\alpha},\delta}$ is the estimated correlation matrix and $\hat{\boldsymbol{c}}_{\hat{\alpha},\delta}$ is the estimated correlation between the locations and the point to predict. Moreover, the associated mean squared error is computed through

$$MSE(\widehat{Z}(\boldsymbol{x}_0)) = \hat{\sigma}^2 (1 - \hat{\boldsymbol{c}}_{\hat{\alpha},\delta}^T \widehat{C}_{\hat{\alpha},\delta}^{-1} \hat{\boldsymbol{c}}_{\hat{\alpha},\delta}). \tag{6}$$

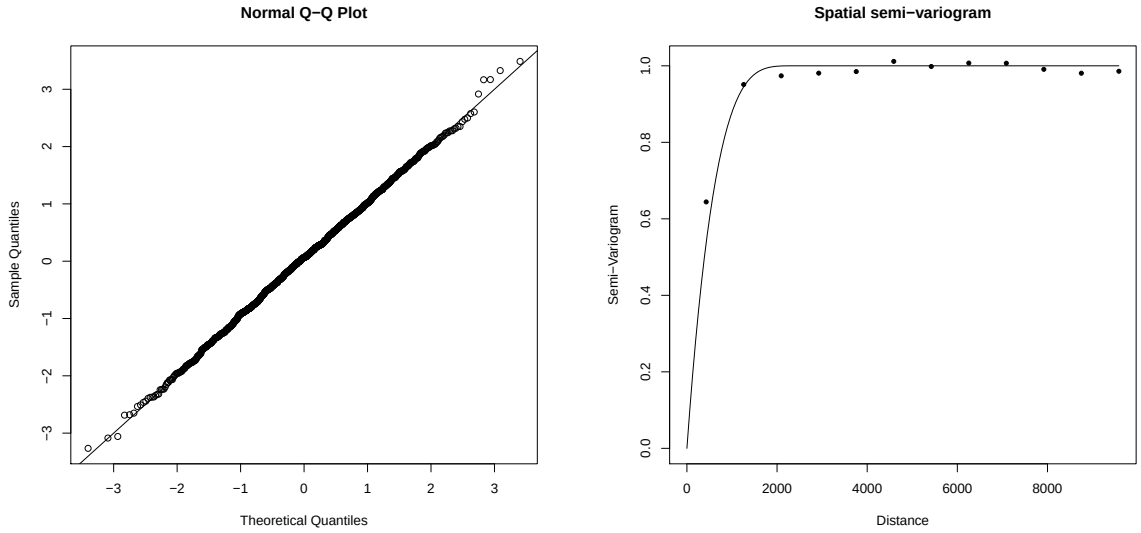

Figure 3: Left: QQ-plot for the model residuals. Right: empirical vs estimated semi-variogram function for the residuals

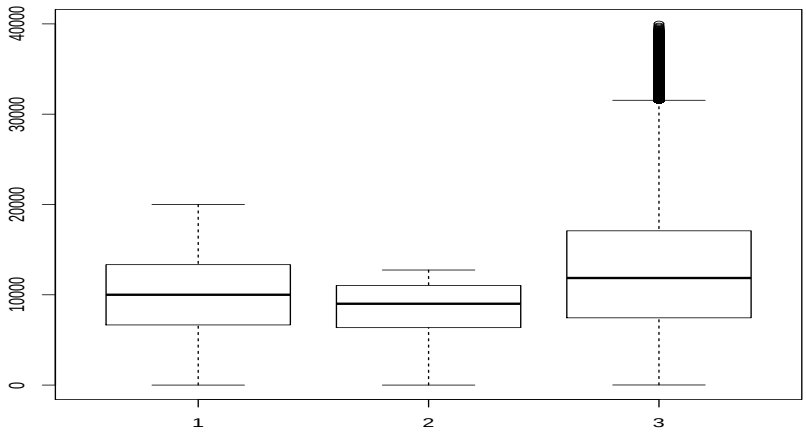

Figure 4: Boxplot of great circle, chordal and euclidean distances

Optimal prediction and associated MSE can be computed using the `GeoKrig` function. Suppose we want to predict our GRF around the north pole area. We need just to specify the spherical location to predict (in lon/lat format):

```
## location to predict given in lon/lat format
loc_to_pred=as.matrix(expand.grid(seq(-180,180,2),seq(-90,90,1)))
```

Then, the optimal predictor (5) and associated MSE (6) (using the maximum likelihood estimated parameters) can be performed with the following code:

```
param_est<-as.list(c(fixed,fit_geo_ml$param))
## computing optimal prediction. and associated MSE
pr_geo=GeoKrig(data=data,loc=loc_to_pred,coordx=coords,
corrmodel=corrmodel,radius=radius,distance="Geod",
param=param_est,sparse=TRUE,mse=TRUE)
```

The matrix of locations to predict, predictions and associated MSE can be obtained as:

```
predictions=cbind(loc_to_pred,pr_geo$pred,pr_geo$mse)
[1,] -180   60 0.13029239 0.6667129
[2,] -178   60 0.12648398 0.7069505
[3,] -176   60 0.11616192 0.7246358
[4,] -174   60 0.09686500 0.7165679
[5,] -172   60 0.06893811 0.6801243
[6,] -170   60 0.03323473 0.6184059
```

and a graphical visualization (Figure 5) of the prediction and associated MSE can be obtained with the following code:

```
globeearth(eye=place("newyorkcity"))
globepoints(loc=loc_to_pred,pch=20,
col =  heat.colors(length(pr_geo$pred),alpha=0.1)[rank(pr_geo$pred)])
##
globeearth(eye=place("newyorkcity"))
globepoints(loc=loc_to_pred,pch=20,
col =  cm.colors(length(pr_geo$mse),alpha=0.1)[rank(pr_geo$mse)])
```

References

Baddeley, A., T. Lawrence, and E. Rubak (2017). *globe: Plot 2D and 3D Views of the Earth, Including Major Coastline*. R package version 1.2-0.

Bevilacqua, M. and C. Gaetan (2015). Comparing composite likelihood methods based on pairs for spatial Gaussian random fields. *Statistics and Computing 25*, 877–892.

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.

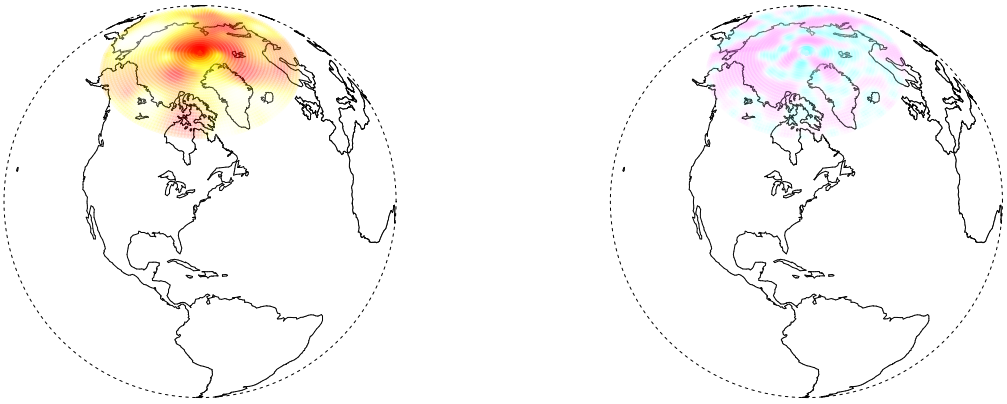

Figure 5: From left to right: prediction and associated MSE in the north pole area

Douglas Nychka, Reinhard Furrer, John Paige, and Stephan Sain (2015). fields: Tools for spatial data. R package version 9.0.

for R by Ray Brownrigg, D. M. P., T. P. Minka, and transition to Plan 9 codebase by Roger Bivand. (2017). *mapproj: Map Projections*. R package version 1.2-5.

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.

Gneiting, T. (2013). Strictly and non-strictly positive definite functions on spheres. *Bernoulli 19*, 1327–1349.

Porcu, E., M. Bevilacqua, and M. Genton (2016). Spatio-temporal covariance and cross covariance functions of the great circle distance on a sphere. *Journal of the American Statistical Association 111*, 888–898.

Robotham, A. (2013). *sphereplot: Spherical plotting*. R package version 1.5.