GeoModels Tutorial: analysis of spatial data with heavy tails using t random fields

Moreno Bevilacqua
Víctor Morales-Oñate

December 18, 2018

Introduction

In this tutorial we show how to analyze spatial data with heavy tails using t random fields (Bevilacqua et al., 2019) with the R package `GeoModels` (Bevilacqua and Morales-Oñate (2018)). The t distribution is a flexible parametric model, which is able to accommodate flexible tail behaviour and, in particular, heavier tails than the ones induced by Gaussian random fields.

We first load the R libraries needed in this tutorial and set the name of the model in the *GeoModels* package. The *GeoModels* package can be loaded in its standard or *OpenCL* version:

```
rm(list=ls())
require(devtools)
install_github("vmoprojs/GeoModels")
#install_github("vmoprojs/GeoModels-OCL")# for OpenCL implementation
require(GeoModels)
require(fields)
require(hypergeo)
require(limma)
model="StudentT"  # model name in the  GeoModels package
set.seed(16)
```

Simulation of t random fields

The definition of a t random field starts by considering a 'parent' Gaussian random field $G = \{G(\boldsymbol{s}), \boldsymbol{s} \in S\}$, where $\boldsymbol{s}$ represents a location in the domain S. In this tutorial we consider $S \subseteq \mathbb{R}^2$. The Gaussian field G is assumed stationary with zero mean, unit variance and correlation function $\rho(\boldsymbol{h}) = \text{cor}(G(\boldsymbol{s}+\boldsymbol{h}), G(\boldsymbol{s}))$.

Given $G_1, \ldots, G_\nu$ independent copies of G, where ν is a positive integer greater than two, let $Y^*_\nu = \{Y^*_\nu(\boldsymbol{s}), \boldsymbol{s} \in S\}$ be a random field defined through a scale mixture:

$$Y^*_\nu(\boldsymbol{s}) = \left(\sum_{i=1}^{\nu} G_i(\boldsymbol{s})^2/\nu\right)^{-\frac{1}{2}} G(\boldsymbol{s}), \tag{1}$$

with marginal distribution t with associated density:

$$f_{Y^*_\nu(\boldsymbol{s})}(y) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\nu\pi}\,\Gamma(\nu/2)} \left(1 + \frac{y^2}{\nu}\right)^{-(\nu+1)/2} \qquad y \in \mathbb{R}. \tag{2}$$

Then $\mathbb{E}(Y_\nu^*(\boldsymbol{s})) = 0$, $\text{var}(Y_\nu^*(\boldsymbol{s})) = \nu/(\nu-2)$ and the correlation function is given by (Bevilacqua et al., 2019):

$$\rho_{Y_\nu^*}(\boldsymbol{h}) = \frac{(\nu-2)\Gamma^2\left(\frac{\nu-1}{2}\right)}{2\Gamma^2\left(\frac{\nu}{2}\right)}\left[{}_2F_1\left(\frac{1}{2},\frac{1}{2};\frac{\nu}{2};\rho^2(\boldsymbol{h})\right)\rho(\boldsymbol{h})\right]. \tag{3}$$

Here ${}_2F_1\left(a,b;c;x\right)$ is the Gaussian hypergeometric function (Abramowitz and Stegun (1970)). In the `GeoModels` package the ${}_2F_1$ function is computed using the function `hypergeo` of the `hypergeo` package (Hankin, 2016).

Then, we define the location-scale transformation process $Y_\nu = \{Y_\nu(\boldsymbol{s}), \boldsymbol{s} \in A\}$ as:

$$Y_\nu(\boldsymbol{s}) := \mu(\boldsymbol{s}) + \sigma Y_\nu^*(\boldsymbol{s}) \tag{4}$$

with $\mathbb{E}(Y_\nu(\boldsymbol{s})) = \mu(\boldsymbol{s})$ and $Var(Y_\nu(\boldsymbol{s})) = \sigma^2\nu/(\nu-2)$ and a spatial regression model can be specified by assuming that $\mu(\boldsymbol{s}) = X(\boldsymbol{s})^T\boldsymbol{\beta}$ where $X(\boldsymbol{s})$ 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 assume $k=2$.

To obtain a simulation from Y_ν we need to specify a regression mean, degrees of freedom and a variance parameters *i.e.* β_1, β_2, ν, σ^2. Moreover we need to specify a parametric correlation $\rho(\boldsymbol{h})$ for the 'parent' Gaussian random field. We first set the spatial coordinates:

```
N=650
coords=cbind(runif(N),runif(N))
plot(coords,pch=20,xlab="",ylab="")
```

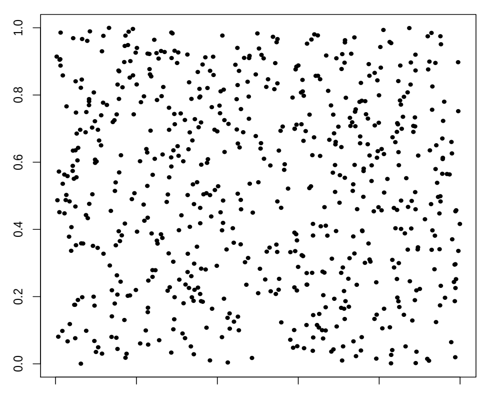

Figure 1: Spatial location sites used in the example.

For the correlation function $\rho(\boldsymbol{h})$ of the 'parent' Gaussian random field G we assume an isotropic Matérn model (Matérn, 1986):

$$\rho_{\alpha,\gamma}(\boldsymbol{h}) = \frac{2^{1-\gamma}}{\Gamma(\gamma)} \left(\|\boldsymbol{h}\|/\alpha\right)^{\gamma} \mathcal{K}_{\gamma}\left(\|\boldsymbol{h}\|/\alpha\right), \qquad \|\boldsymbol{h}\| \geq 0. \tag{5}$$

where $\mathcal{K}_\gamma$ is a modified Bessel function of the second kind of order γ, $\gamma > 0$ is the smoothness parameter and $\alpha > 0$ the spatial scale parameter. Then, we set the parameter associated to this correlation model:

```
corrmodel = "Matern"          ## correlation model
scale = 0.2/3                 ## scale parameter
smooth=0.5                    ## smooth parameter
nugget=0         # nugget parameter
```

and we set the degrees of freedom and variance parameters of the t random field:

```
df = 5          # degrees of freedom
sill= 1         # variance parameter
```

Finally we set the mean regression parameters and the regression matrix:

```
mean = 0.5; mean1= -1   # regression paramteres
a0=rep(1,N);a1=runif(N,-1,1)
X=cbind(a0,a1)     ## regression matrix
```

We are now ready to simulate a realization of the t random field Y_ν using the function *GeoSim*. Simulation is performed exploiting the stochastic representation (1), where the Gaussian fields involved are generated through Cholesky decomposition:

```
param=list(nugget=nugget,mean=mean,mean1=mean1, scale=scale,
               smooth=smooth, sill=sill,df=1/df)
data <- GeoSim(coordx=coords,corrmodel=corrmodel,
                param=param,model=model,X=X)$data
```

Note that the parametrization in the package *GeoModels* uses the inverse of the degrees of freedom as suggested in Bevilacqua et al. (2019).

Estimation of t random fields

Estimation of regression, degrees of freedom and correlation parameters of the t random field Y_ν can be performed using pairwise likelihood estimation. Let $f_{\boldsymbol{Y}^*_{\nu;ij}}(y_i, y_j)$ the density of the bivariate random vector $(Y_\nu^*(\boldsymbol{s}_i), Y_\nu^*(\boldsymbol{s}_j))^T$ given by (Bevilacqua et al., 2019):

$$\begin{aligned} f_{\boldsymbol{Y}^*_{\nu;ij}}(y_i, y_j) &= \frac{\nu^\nu l_{ij}^{-\frac{(\nu+1)}{2}} \Gamma^2\left(\frac{\nu+1}{2}\right)}{\pi\Gamma^2\left(\frac{\nu}{2}\right)(1-\rho^2(\boldsymbol{h}))^{-(\nu+1)/2}} F_4\left(\frac{\nu+1}{2}, \frac{\nu+1}{2}, \frac{1}{2}, \frac{\nu}{2}; \frac{\rho^2(\boldsymbol{h})y_i^2y_j^2}{l_{ij}}, \frac{\nu^2\rho^2(\boldsymbol{h})}{l_{ij}}\right) \\ &+ \frac{\rho(\boldsymbol{h})y_iy_j\nu^{\nu+2} l_{ij}^{-\frac{\nu}{2}-1}}{2\pi(1-\rho^2(\boldsymbol{h}))^{-\frac{(\nu+1)}{2}}} F_4\left(\frac{\nu}{2}+1, \frac{\nu}{2}+1, \frac{3}{2}, \frac{\nu}{2}; \frac{\rho^2(\boldsymbol{h})y_i^2y_j^2}{l_{ij}}, \frac{\nu^2\rho^2(\boldsymbol{h})}{l_{ij}}\right) \end{aligned} \tag{6}$$

where $l_{ij} = [(y_i^2+\nu)(y_j^2+\nu)]$ and

$$F_4(a, b; c, c'; w, z) = \sum_{k=0}^{\infty}\sum_{m=0}^{\infty} \frac{(a)_{k+m}(b)_{k+m} w^k z^m}{k!m!(c)_k(c')_m}, \quad |\sqrt{w}| + |\sqrt{z}| < 1.$$

is the Appell function of the fourth type (Gradshteyn and Ryzhik, 2007).

Given a partial realization $(y(\boldsymbol{s}_1), \dots, y(\boldsymbol{s}_N))^T$ of the t random process Y_ν defined in equation (4), the density of the bivariate random vector $(Y_\nu(\boldsymbol{s}_i), Y_\nu(\boldsymbol{s}_j))^T$ can be obtained from (6) as:

$$f_{\boldsymbol{Y}_{\nu;ij}}(y(\boldsymbol{s}_i), y(\boldsymbol{s}_j)) = \frac{1}{\sigma^2} f_{\boldsymbol{Y}^*_{\nu;ij}}\left(\frac{y(\boldsymbol{s}_i)-\mu(\boldsymbol{s}_i)}{\sigma}, \frac{y(\boldsymbol{s}_j)-\mu(\boldsymbol{s}_j)}{\sigma}\right). \tag{7}$$

Then, the pairwise likelihood function is defined as:

$$pl(\boldsymbol{\theta}) = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} log(f_{\boldsymbol{Y}_{\nu;ij}}(y(\boldsymbol{s}_i), y(\boldsymbol{s}_j)))w_{ij} \tag{8}$$

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

$$w_{ij} := \begin{cases} 1 & ||\boldsymbol{s}_i - \boldsymbol{s}_j|| < d \\ 0 & \text{otherwise} \end{cases}. \tag{9}$$

and in this case $\boldsymbol{\theta} = (\beta_1, \beta_2, \nu, \sigma^2, \alpha, \delta)^T$. The pairwise likelihood estimator $\hat{\boldsymbol{\theta}}_{pl}$ is obtained maximizing (8) with respect to $\boldsymbol{\theta}$. In the *GeoModels* package, we can choose the fixed parameters and the parameters that must be estimated.

As argued in Bevilacqua et al. (2019), the degrees of freedom must be fixed to a positive integer value greater than two (in some special cases $\nu > 2$ without any restriction on degrees of freedom parameter).

If we assume ν unknown, the degrees of freedom can be fixed trough a two-step estimation. In the first step, we estimate the parameters, including ν without any restriction on its parametric space. Pairwise likelihood estimation is performed using the function *GeoFit*:

```
fixed1<-list(nugget=nugget,smooth=smooth)
start1<-list(mean=mean, mean1=mean1,scale=scale,sill=sill,df=1/df)
fit2 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
         maxdist=0.04,X=X,start=start1,fixed=fixed1, model = model)
         #GPU=0,local=c(1,1))
```

Note that the option *maxdist=0.04* set the compact support of the weight function (9) i.e. $d = 0.04$. Then, we round the estimation of ν obtained at first step:

```
DF=as.numeric(round(1/fit2$param["df"]))
if(DF==2) DF=3
print(DF)
[1] 5
```

In this case, the rounded estimated value of ν matches the true vale of ν. Finally, we perform the second step estimation keeping fixed the degrees of freedom:

```
start<-list(mean=mean, mean1=mean1,scale=scale,sill=sill)
fixed<-list(nugget=nugget,df=1/DF,smooth=smooth)
fit2 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
  maxdist=0.04,X=X,start=start,fixed=fixed, model = model)
  # GPU=0,local=c(1,1))
```

In the case that the OpenCL version of the `GeoModels` package was loaded, the option `GPU=0,local=c(1,1)` allows to speed up the computation performance. The first parameter (`GPU=0`) sets the computing device. You can call the available devices in your computer through `DeviceInfo()` and choose the associated number. The second argument (`local=c(1,1)`) lets you set the number of local work-items of the OpenCL setup.

The object `fit2` include informations about the pairwise likelihood estimation:

```
fit2
##################################################################
Maximum Composite-Likelihood Fitting of StudentT Random Fields
Setting: Marginal Composite-Likelihood
Model associated to the likelihood objects: StudentT
Type of the likelihood objects: Pairwise
Covariance model: Matern
Number of spatial coordinates: 650
Number of dependent temporal realisations: 1
```

```
Type of the random field: univariate
Number of estimated parameters: 4
Type of convergence: Successful
Maximum log-Composite-Likelihood value: -2995.78
Estimated parameters:
    mean      mean1      scale       sill
 0.15652   -0.93409   0.06795   1.08181
#################################################################
```

Checking model assumptions

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

$$\widehat{Y_\nu^*(\boldsymbol{s}_i)} = \frac{y(\boldsymbol{s}_i) - X(\boldsymbol{s}_i)^T \widehat{\boldsymbol{\beta}}}{(\widehat{\sigma}^2)^{\frac{1}{2}}} \quad i = 1, \dots N$$

can be viewed as a realization of the process Y_ν^*. The residuals can be computed using the *GeoResiduals* function:

```
res=GeoResiduals(fit2)  # computing residuals
```

The marginal distribution assumption on the residuals can be graphically checked with a qq-plot using the *qqt* function in the *R* package *limma*:

```
### checking model  residuals assumptions: marginal distribution
qqt(res$data,df=DF)
abline(0,1)
```

The covariance model assumption can be checked comparing the empirical and the estimated semi-variogram using the *GeoVariogram* and *GeoCovariogram* functions:

```
### checking model  residuals assumptions: covariance model
vario <- GeoVariogram(data=res$data,coordx=coords,maxdist=0.4)
GeoCovariogram(res,show.vario=TRUE, vario=vario,pch=20)
```

The semi-variogram is computed using the correlation function (3).

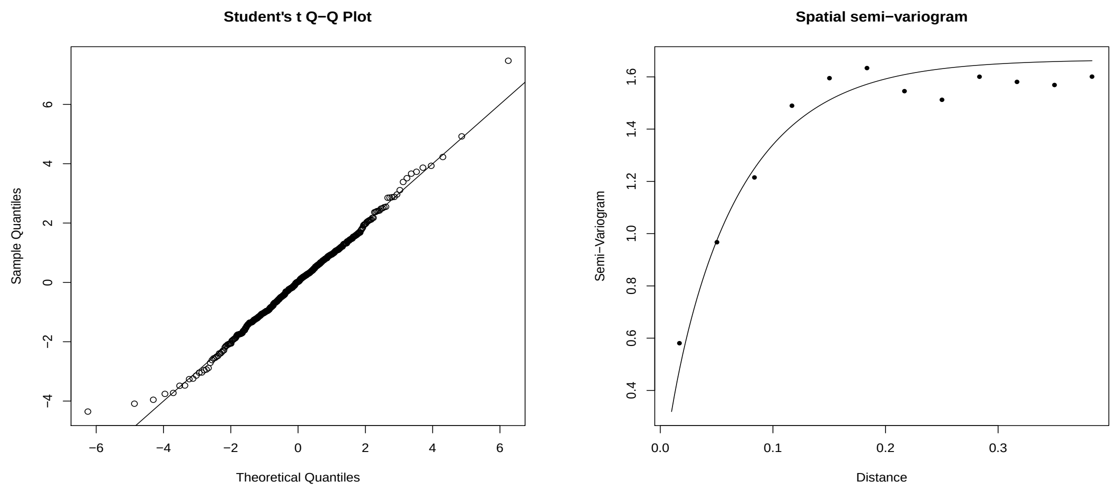

Figure 2: From left to right: qq-plot of the residuals using the t distribution and empirical vs estimated semi-variogram for the residuals.

Prediction of t random fields

For a given spatial location $\boldsymbol{s}_0$ with associated covariates $X(\boldsymbol{s}_0)$, the optimal linear prediction (assuming known the parameters) of a t random field is given by:

$$\widehat{Y}_\nu(\boldsymbol{s}_0) = X(\boldsymbol{s}_0)^T\boldsymbol{\beta} + \sum_{i=1}^{N} \lambda_i[y(\boldsymbol{s}_i) - X(\boldsymbol{s}_i)^T\boldsymbol{\beta}] \tag{10}$$

where the vector of weights $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_N)'$ is given by $\boldsymbol{\lambda} = R_\nu^{-1}\boldsymbol{c}_\nu$ and

- $\boldsymbol{c}_\nu = (cor(Y_\nu(\boldsymbol{s}_0), Y_\nu(\boldsymbol{s}_1)), \dots, cor(Y_\nu(\boldsymbol{s}_0), Y_\nu(\boldsymbol{s}_N)))^T$.
- $R_\nu = [\text{cor}(Y_\nu(\boldsymbol{s}_i), Y_\nu(\boldsymbol{s}_j)]]_{i,j=1}^N$ is the correlation matrix.

Moreover the associated mean square error (MSE) is given by:

$$MSE(\widehat{Y}_\nu(\boldsymbol{s}_0)) = \sigma^2(1 - \boldsymbol{c}_\nu^T R_\nu^{-1}\boldsymbol{c}_\nu). \tag{11}$$

If the parameters are unknown, both (10) and (11) can be computed replacing the parameters with the pairwise likelihood estimates. In particular, R_ν and $\boldsymbol{c}_\nu$ can be computed using (3), the estimates of the Matérn correlation function in $\rho(\boldsymbol{h})$ and of the degrees of freedom.

Kriging and associated MSE can be obtained using the `GeoKrig` function. We first need to specify the spatial locations to predict and, in this example, we consider a spatial regular grid:

```
xx=seq(0,1,0.015)
loc_to_pred=as.matrix(expand.grid(xx,xx))
Nloc=nrow(loc_to_pred)
Xloc=cbind(rep(1,Nloc),runif(Nloc))
```

Then the optimal linear prediction (10), using the estimated parameters, can be performed using the `GeoKrig` function:

```
param_est=as.list(c(fit2$param,fixed))
pr=GeoKrig(data=data, coordx=coords,loc=loc_to_pred, X=X,Xloc=Xloc,
     corrmodel=corrmodel,model=model,mse=TRUE,param= param_est)
```

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

```
par(mfrow=c(1,3))
colour = rainbow(100)
#### map of  data
quilt.plot(coords[,1], coords[,2], data,col=colour,main="Data")
# linear kriging
map=matrix(pr$pred,ncol=length(xx))
image.plot(xx,xx,map,col=colour,xlab="",ylab="",main="SimpleKriging")
#associated mean squared error
map_mse=matrix(pr$mse,ncol=length(xx))
image.plot(xx,xx,map_mse,col=colour,xlab="",ylab="",main="MSE")
```

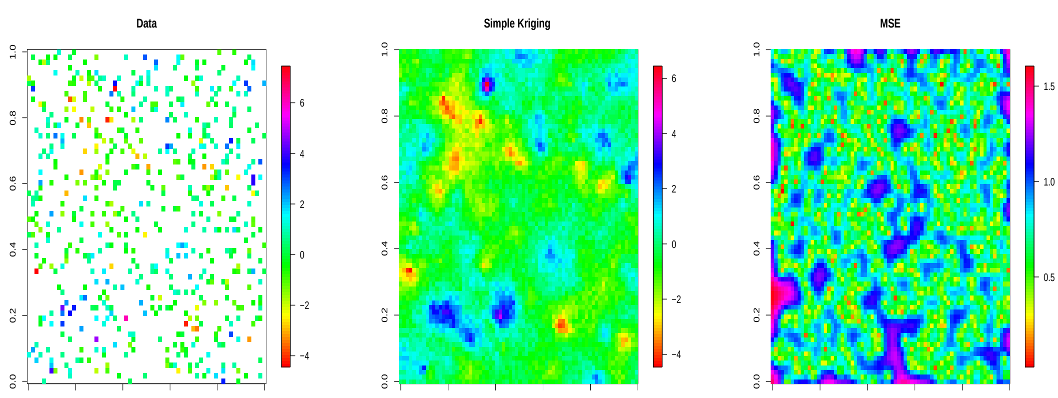

Figure 3: From left to right: observed spatial data, associated kriging map and mean square error map.

References

Abramowitz, M. and I. A. Stegun (Eds.) (1970). *Handbook of Mathematical Functions*. New York: Dover.

Bevilacqua, M., C. Caamaño, R. B. A. Valle, and V. Morales-Oñate (2019). On spatial (skew) t processes and applications. *ArXiv e-prints*.

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.

Gradshteyn, I. and I. Ryzhik (2007). *Table of Integrals, Series, and Products* (eight ed.). Cambridge, MA: Academic Press.

Hankin, R. K. S. (2016). *hypergeo: The Gauss Hypergeometric Function*. R package version 1.2-13.

Matérn, B. (1986). *Spatial Variation: Stochastic Models and their Applications to Some Problems in Forest Surveys and Other Sampling Investigations* (2nd ed.). Heidelberg: Springer.