`GeoModels` Tutorial: simulation, estimation and prediction of spatial data with heavy tails using t random fields

Moreno Bevilacqua
Christian Caamaño-Carrillo
Víctor Morales-Oñate

May 11, 2022

Introduction

In this tutorial we show how to analyze spatial data with heavy tails using t random fields (Bevilacqua et al., 2020) with the R package `GeoModels` (Bevilacqua et al. (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.

```
rm(list=ls());
require(devtools);
require(GeoModels);
require(fields);
require(hypergeo);
require(limma);
model="StudentT";  # model name in the  GeoModels package
set.seed(199);
```

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 the spatial case *i.e.* $S \subseteq \mathbb{R}^2$. However, the package `GeoModels` allows to work also with spatio-temporal data or data defined on a sphere of arbitrary radius. The Gaussian field G is assumed weakly stationary with zero mean, unit variance and correlation function $\rho(\boldsymbol{h}) = \mathrm{cor}(G(\boldsymbol{s}+\boldsymbol{h}), G(\boldsymbol{s}))$.

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 $\mathbb{E}(Y_\nu^*(\boldsymbol{s})) = 0$, $\text{var}(Y_\nu^*(\boldsymbol{s})) = \nu/(\nu-2)$, $\nu > 2$ and the correlation function is given by (Bevilacqua et al., 2020):

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

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 $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)^T$ 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 regression mean, degrees of freedom and variance parameters *i.e.* β_1, β_2, ν, σ^2 respectively. Additionally, we need to specify a parametric correlation $\rho(\boldsymbol{h})$ for the 'parent' Gaussian random field. We first set the spatial coordinates:

```
N=600;
coords=cbind(runif(N),runif(N));
plot(coords ,pch=20,xlab="",ylab="");
```

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

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

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
```

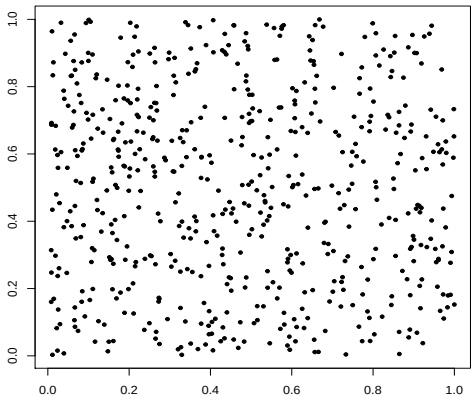

Figure 1: Spatial location sites used in the tutorial.

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. (2020).

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., 2020):

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

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 $(y(\boldsymbol{s}_1), \ldots, y(\boldsymbol{s}_N)^T$ 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 $\boldsymbol{\theta}$. An efficient way to specify the weights from computational and efficient viewpoint is based on distances:

$$w_{ij}(k) = \begin{cases} 1 & ||\mathbf{s}_i - \mathbf{s}_j|| < k \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Here $k > 0$ is an arbitrary positive distance 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 can be estimated.

As argued in Bevilacqua et al. (2020), 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. Under the reparametrization of the degrees of freedom the parametric space is $(0, 0.5)$. Pairwise likelihood estimation can be performed

using the function *GeoFit*. In this example, we perform optimization of (8) using the function `nlminb` that allows box-constrained optimization using PORT routines. However other type of optimization algorithms available in *R* can be used (`BFGS` or `Nelder-Mead` for instance).

```
optimizer="nlminb";
fixed1<-list(nugget=nugget,smooth=smooth);
start1<-list(mean=mean, mean1=mean1,scale=scale,sill=sill,df=1/df);
I=Inf;
lower1<-list(mean=-I, mean1=-I,scale=0,sill=0,df=0);
upper1<-list(mean=I, mean1=I,scale=I,sill=I,df=0.5);
fit1 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
optimizer=optimizer,lower=lower1,upper=upper1,
likelihood="Marginal",type="Pairwise",
maxdist=0.04,X=X,start=start1,fixed=fixed1, model = model)
```

Note that the option $maxdist = 0.04$ set the distance k in the weight function (9) i.e. $k = 0.05$. To guarantee the existence of the t process we need to round the estimation of ν obtained at first step:

```
DF=as.numeric(1/round(fit1$param["df"]));
print(DF);
[1] 5
```

In this case, the rounded estimated value of ν matches the true vale of ν. Then, 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)
lower<-list(mean=-I, mean1=-I,scale=0,sill=0)
upper<-list(mean=I, mean1=I,scale=I,sill=I)
fit2 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
optimizer=optimizer, lower=lower,upper=upper,
likelihood="Marginal",type="Pairwise",
maxdist=0.04,X=X,start=start,fixed=fixed, model = model)
```

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

```
fit2
##########################################################################
```

```
Maximum  Composite-Likelihood Fitting of StudentT Random Fields
Setting: Marginal Composite-Likelihood
Model: StudentT
Type of the likelihood objects: Pairwise
Covariance model: Matern
Optimizer: nlminb
Number of spatial coordinates: 600
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: -2415.88
Estimated parameters:
    mean     mean1     scale      sill
 0.48132  -0.97126   0.06188   0.82063
###################################################################
```

An alternative, less efficient and computationally easier estimator can be obtained by assuming a misspecified Gaussian model in the pairwise likelihood estimation method. Specifically, if in the estimation step we assume a Gaussian random field with the same mean, variance and correlation function of the t random field (see Bevilacqua et al. (2020)), then a weighted misspecified Gaussian pairwise likelihood estimation can be performed changing the name of the model in the function `GeoFit`:

```
fit3 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
 optimizer="nlminb", lower=lower,upper=upper,maxdist=0.04,
 likelihood="Marginal",type="Pairwise",
 X=X,start=start,fixed=fixed, model = "Gaussian_misp_StudentT")
```

The two estimates are quite similar in this case but in general the misspecified Gaussian assumption leads to a loss of efficiency that increase when degreasing the degrees of freedom. (see Bevilacqua et al. (2020) for a comparison between the two estimators).

```
fit2$param; fit3$param
       mean        mean1       scale        sill
 0.48131540 -0.97126017  0.06187684  0.82062641
       mean        mean1       scale        sill
 0.53194054 -0.92580720  0.06431009  0.80292932
```

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 trough a qq-plot using the *GeoQQ* function (see Figure 2, left part):

```
### checking model  residuals assumptions: marginal distribution
GeoQQ(res)
```

The covariance model assumption can be checked comparing the empirical and the estimated semi-variogram using the *GeoVariogram* and *GeoCovariogram* functions (see Figure 2, right part):

```
### 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);
```

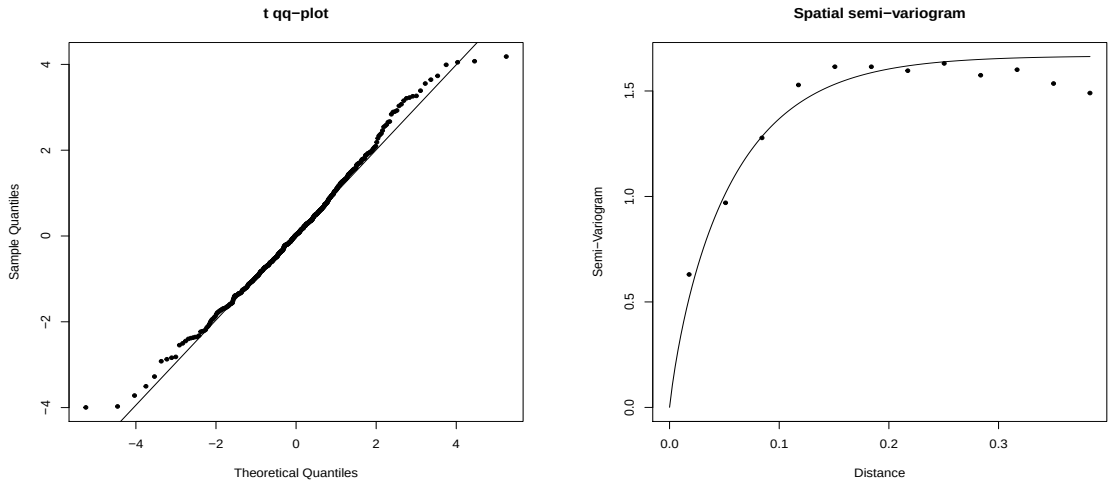

Figure 2: From left to right: qq-plot of the residuals using the t_5 distribution and empirical vs estimated semi-variogram for the residuals.

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

Prediction of t random fields

For a given spatial location $\boldsymbol{s}_0$ with associated covariates $X(\boldsymbol{s}_0)$, the optimal linear prediction 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 $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)'$ is given by $\boldsymbol{\lambda} = R_\nu^{-1}\boldsymbol{c}_\nu$ and

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

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

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

The predictor can be viewed as an optimal Gaussian predictor assuming (3) as correlation function. 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) coupled with the estimates of the Matérn correlation function 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(0,1,0.012);
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 (computation can be time consuming):

```
param_est=as.list(c(fit1$param,fixed1));
pr=GeoKrig(data=data, coordx=coords,loc=loc_to_pred, X=X,Xloc=Xloc,
     corrmodel=corrmodel,model=model,mse=TRUE,param= param_est)
```

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

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

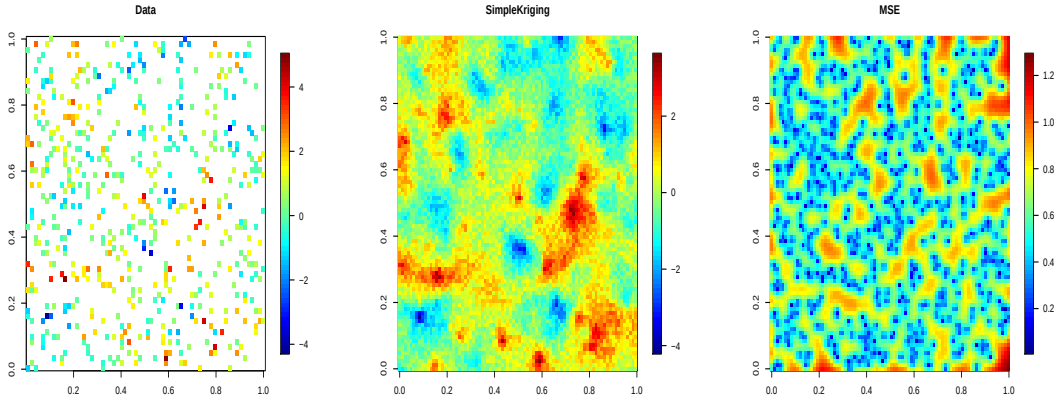

Figure 3: From left to right: observed spatial data, associated kriging map and mean square error map.

References

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

Bevilacqua, M., C. Caamaño-Carrillo, R. B. Arellano-Valle, and V. Morales-Oñate (2020). Non-gaussian geostatistical modeling using (skew) t processes. *Scandinavian Journal of Statistics*, 1–34.

Bevilacqua, M., V. Morales-Oñate, and C. Caamaño-Carrillo (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.