`GeoModels` Tutorial: analysis of temperature in Australia using t random fields

Moreno Bevilacqua
Christian Caamaño

August 14, 2023

Introduction

In this tutorial we show how to analyze a data set of temperature data observed in Australia with the R package `GeoModels` (Bevilacqua et al. (2018)). The data has been analyzed in Bevilacqua et al. (2020) where a flexible random field with t marginal distribution has been proposed. We first load the R libraries needed in this tutorial:

```
library(GeoModels)
library(fields)
require(limma)
require(oz)
library(maps)
require(maptools)
require(mapdata)
require(geoR)
```

Preliminary data analysis

The dataset is a subset of a global data set of merged maximum daily temperature measurements from the Global Surface Summary of Day data (GSOD) with European Climate Assessment & Dataset (ECA&D) data in July 2011. We first import the data that can be found in the package `GeoModels`

```
data(austemp)
head(austemp)
         lon     lat temp        X
[1,] 130.883 -12.417 31.0 23.11499
[2,] 134.333 -13.667 28.8 22.31359
[3,] 145.317 -14.967 25.8 21.46886
[4,] 131.017 -16.400 31.3 20.52493
[5,] 143.533 -18.300 26.8 19.25369
[6,] 115.017 -21.450 25.3 17.10014
```

Here, X is a covariate called *geometric temperature* which represents the geometric position of a particular location on Earth and the day of the year (Kilibarda et al., 2014) and *temp* is the maximum temperature. A linear relation between the *geometric temperature* and the maximum temperature can be appreciated from Figure 1.

```
coords=cbind(austemp[,1],austemp[,2])
temp=austemp[,3]
plot(temp,austemp[,4],pch=20,xlab="Geom. temperature",ylab="Max. temperature")
```

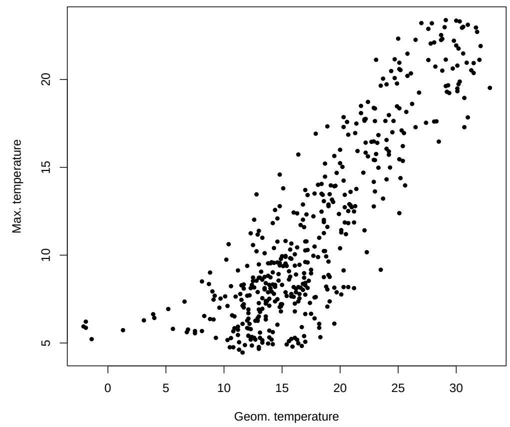

Figure 1: *Geometric temperature* versus maximum temperature data.

Spatial coordinates are given in longitude and latitude expressed as decimal degrees and in this tutorial we consider random fields defined on the planet Earth sphere approximation $\mathbb{S}^2 = \{\boldsymbol{s} \in \mathbb{R}^3 : ||\boldsymbol{s}|| = 6371\}$ using the geodesic distance . For this reason we set

```
radius=6371
distance="Geod"
```

Here 6371 is the radius of the earth expressed in Km. The subset we consider is depicted in Figure 2 and consists of the maximum temperature observed on July 5 in 446 location sites in the region with longitude $[110, 154]$ and latitude $[-39, -12]$.

```
quilt.plot(coords,temp,xlab="long",ylab="lat")
oz(states=FALSE,add=T, lwd=2)
```

The marginal distribution of the data (see the histogram in Figure 3 left part) suggests that a Gaussian random field assumption could not be a reasonable model.

```
hist(temp,main="Histogram of Maximum temperature",nclass=13)
```

A random field with heavy tails marginal distribution could be more appropriate in particular when modeling the left tail. Additionally the h-scatterplot obtained using the function

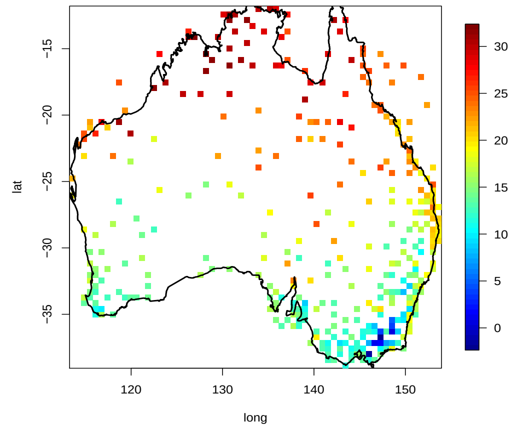

Figure 2: Coloured map of observed maximum temperature.

GeoScatterplot indicate non-elliptical dependence for the bivariate distributions (see Figure, 3 left part).

```
GeoScatterplot(data=temp,coordx=coords,distance="Geod",maxdist=500,
                    numbins=6)
```

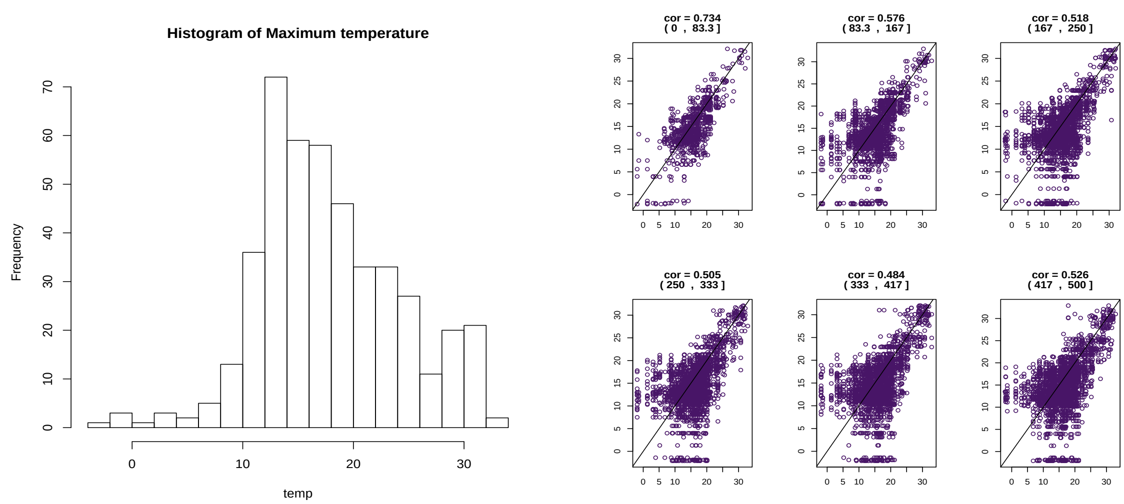

Figure 3: From left to right: histrogram of temperature data and associated $h-$scatterlplot.

Finally, the empirical semi-variogram in Figure 4, highlights spatial dependence with a practical range of 300 KM approximatively. Additionally, the nugget effect is negligible.

```
maxdist=max(rdist.earth(coords,miles=F,R=radius))
vario<-GeoVariogram(coordx=coords,data=temp,distance="Geod",
```

```
                        maxdist=maxdist/5,radius=radius)
plot(vario, xlab='GC␣distance', ylab=expression(gamma(GC)),
         cex=1.5,cex.lab=1.5,
ylim=c(0, max(vario$variograms)), pch=20,
main="Semi-variogram")
##########################################
```

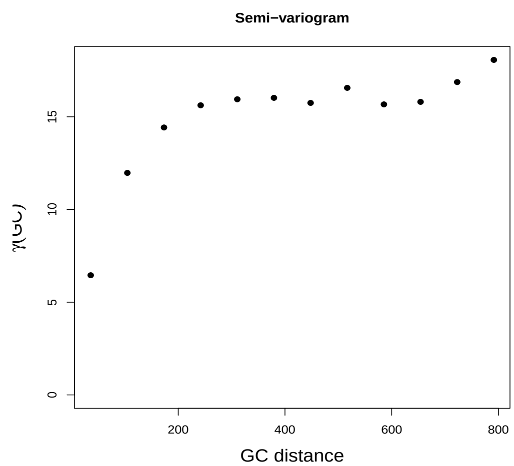

Figure 4: Empirical semi-variogram of temperature data.

This preliminary analysis suggest that a t random field as proposed in Bevilacqua et al. (2020) could be a suitable model for the Australian maximum temperature data. In the next Sections we describe how to estimate and predict the t random fields proposed in Bevilacqua et al. (2018) using the package `GeoModels`.

1 Gaussian and t random fields

Let $\mathbb{S}^2$ the sphere of $\mathbb{R}^3$ of radius $R = 6371$ defined as $\mathbb{S}^2 = \{\boldsymbol{x} \in \mathbb{R}^3 : ||\boldsymbol{x}|| = 6371\}$. We first consider a zero mean, unit variance standard Gaussian random field $G^* = \{G^*(\boldsymbol{s}), \boldsymbol{s} \in \mathbb{S}^2\}$ with geodesically isotropic Matérn function (Gneiting, 2013) that is:

$$\rho(d_{GC}) = \mathcal{M}_{\alpha,\psi}(d_{GC}) = \frac{2^{1-\psi}}{\Gamma(\psi)} \left(d_{GC}/\alpha\right)^{\psi} \mathcal{K}_{\psi}\left(d_{GC}/\alpha\right), \qquad d_{GC} \geq 0, \tag{1}$$

where d_{GC} is the geodesic distance and $\mathcal{K}_{\psi}$ is a modified Bessel function of the second kind of order ψ. Additionally, $\alpha > 0$ and $0 < \psi \leq 0.5$ guarantee the positive definiteness of the model in $\mathbb{S}^2$. In particular when $\psi = 0.5$ then (1) reduced to the exponential correlation model

$$\mathcal{M}_{\alpha,0.5}(d_{GC}) = e^{-d_{GC}/\alpha}. \tag{2}$$

Hereafter, we work with the exponential correlation model.

We consider two RFs in our analysis. The first is a location-scale transformation of G^* that is a RF $G = \{G(\boldsymbol{s}), \boldsymbol{s} \in \mathbb{S}^2\}$ defined as:

$$G(\boldsymbol{s}) := \mu(\boldsymbol{s}) + \sigma G^*(\boldsymbol{s}) \tag{3}$$

with $\mathbb{E}(G(\boldsymbol{s})) = \mu(\boldsymbol{s}) \in \mathbb{R}$ and $Var(G(\boldsymbol{s})) = \sigma^2 \in R^+$ where $\mu(\boldsymbol{s})$ is the spatial mean function. We now briefly resume the t random fields as proposed in Bevilacqua et al. (2020). Given $G_1^*, \ldots, G_\nu^*$ independent copies of G^*, where ν is a positive integer greater than two, let $Y_\nu^* = \{Y_\nu^*(\boldsymbol{s}), \boldsymbol{s} \in \mathbb{S}^2\}$ be a random field defined through a scale mixture:

$$Y_\nu^*(\boldsymbol{s}) = \left(\sum_{i=1}^{\nu} G_i^*(\boldsymbol{s})^2/\nu\right)^{-\frac{1}{2}} G^*(\boldsymbol{s}), \tag{4}$$

with t_ν marginal distribution with associated density:

$$f_{Y_\nu^*(\boldsymbol{s})}(y) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\nu\pi}\,\Gamma(\nu/2)} \left(1 + \frac{y^2}{\nu}\right)^{-(\nu+1)/2} \qquad y \in \mathbb{R}. \tag{5}$$

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

$$\rho_{Y_\nu^*}(d_{GC}) = \frac{(\nu-2)\Gamma^2\left(\frac{\nu-1}{2}\right)}{2\Gamma^2\left(\frac{\nu}{2}\right)} \left[{}_2F_1\left(\frac{1}{2}, \frac{1}{2}; \frac{\nu}{2}; \rho^2(\boldsymbol{h})\right) \rho(d_{GC})\right]. \tag{6}$$

where $\rho(d_{GC})$ is the correlation function in (2) and ${}_2F_1(a, b; c; x)$ 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) and using the `hyp2f1` c code function in the SciPy Python library.

Then, we define the location-scale transformation process $Y_\nu = \{Y_\nu(\boldsymbol{s}), \in \mathbb{S}^2\}$ as:

$$Y_\nu(\boldsymbol{s}) := \mu(\boldsymbol{s}) + \sigma Y_\nu^*(\boldsymbol{s}) \tag{7}$$

with $\mathbb{E}(Y_\nu(\boldsymbol{s})) = \mu(\boldsymbol{s})$ and $Var(Y_\nu(\boldsymbol{s})) = \sigma^2\nu/(\nu-2)$ and

For both RFs we assume a regression model for the spatial mean $\mu(\boldsymbol{s}) = X(\boldsymbol{s})^T\boldsymbol{\beta}$ with $\boldsymbol{\beta} = (\beta_1, \beta_2)^T$ and $X(\boldsymbol{s}) = (1, M(\mathbf{s}))^T$ where $M(\mathbf{s})$ is the geometric temperature covariate in the Australian maximum temperature dataset. To obtain the names of the correlation parameters for the correlation model and the names of the nuisance parameters for the Gaussian and t models, two useful functions are `CorrParam` and `NuisParam`.

```
corrmodel="Matern"
CorrParam(corrmodel)
[1] "scale"  "smooth"
NuisParam("Gaussian",num_betas=2)
[1] "mean"   "mean1"  "nugget" "sill"
NuisParam("StudentT",num_betas=2)
[1] "mean"   "mean1"  "df"     "nugget" "sill"
NN=nrow(coords)
X=cbind(rep(1,NN),austemp[,4]) # matrix covariates
```

Here `nugget` is the τ^2 parameter, `sill` is the σ^2 parameter, `df` is the ν parameter and `mean`, `mean1` are the β_1 and β_2 parameters respectively. For the special case of the Matérn model in equation (1) `scale`, `smooth` are the α and ψ parameters respectively.

Estimation of temperature data

Given a realization $\boldsymbol{G} = (g(\boldsymbol{s}_1), g(\boldsymbol{s}_2), \ldots, g(\boldsymbol{s}_N))^T$, with $N = 446$, from the Gaussian random field G in equation (3) and correlation model (2) the estimation of the parameters can be performed using maximum likelihood method that is maximizing the Gaussian multivariate pdf

$$f_{\boldsymbol{G}}(g_1, \ldots, g_N; \boldsymbol{\theta}_G) = (2\pi)^{-N/2}|\sigma^2 R|^{-1/2} \exp\left\{-\frac{(\boldsymbol{G} - X\boldsymbol{\beta})^T R^{-1}(\boldsymbol{G} - X\boldsymbol{\beta})}{2\sigma^2}\right\} \tag{8}$$

with respect to $\boldsymbol{\theta}_G = (\beta_1, \beta_2, \sigma^2, \alpha)^T$ where R is the correlation matrix associated with the correlation model (2).

However, in this tutorial we focus on a estimation method called weighted pairwise likelihood Bevilacqua and Gaetan (2015) that involves only the pdf of the generic random pair $\boldsymbol{G}_{ij} = (G(\boldsymbol{s}_i), G(\boldsymbol{s}_j))$, that is an estimator obtained maximizing the function:

$$wpl(\boldsymbol{\theta}_G) = \sum_{i=i}^{N} \sum_{j=1, j\neq i}^{N} log(f_{\boldsymbol{G}_{ij}}(g_i, g_j))w_{ij} \tag{9}$$

where w_{ij} are non-negative weights, not depending on $\boldsymbol{\theta}_G$.

An efficient way to specify the (non symmetric) weights from computational and efficient viewpoint is based on neighborhoods:

$$w_{ij}(k) = \begin{cases} 1 & \mathbf{s}_i \in N_k(\mathbf{s}_j) \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where $N_k(\mathbf{s}_l)$ is the set of the neighbors of order $k = 1, 2, \dots$ of the point $\mathbf{s}_l$.

Similarly, given a realization $\boldsymbol{Y}_\nu = (y(\boldsymbol{s}_1), \dots, y(\boldsymbol{s}_N))^T$ of the t random field Y_ν defined in equation (7), the pairwise likelihood function associated to Y_ν is given by

$$pl(\boldsymbol{\theta}_{Y_\nu}) = \sum_{i=i}^{N} \sum_{j=i,j\neq i}^{N} log(f_{\boldsymbol{Y}_{\nu;ij}}(y_i, y_j))w_{ij} \tag{11}$$

where $\boldsymbol{\theta}_{Y_\nu} = (\beta_1, \beta_2, \sigma^2, \alpha, \nu)^T$. The pdf of the bivariate distribution of the t random field is given in Bevilacqua et al. (2020) and it depends on the Appell function double series of the fourth type (Gradshteyn and Ryzhik, 2007).

The weighted pairwise likelihood estimators of the Gaussian and t_ν random fields are obtained maximizing (9) and (11) with respect to $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_{Y_\nu}$ respectively.

In the `GeoModels` package, pairwise likelihood estimation can be performed using the function `GeoFit2`. We perform optimization of (9) and (11) using the optimization algorithm `nlminb` that allows box-constrained optimization. We use the following code to estimate the parameters $\boldsymbol{\theta}_G$ of the Gaussian random field with weighted pairwise likelihood: Note that we can choose the fixed parameters and the parameters that must be estimated.

```
mean=7.5
mean1=1
sill=8; nugget=0
smooth=0.5;scale=60;
optimizer="nlminb"
fixed1=list(nugget=0,smooth=smooth)
I=Inf
start1=list(mean=mean,mean1=mean1,scale=scale,sill=sill)
lower1=list(mean=-I,mean1=-I,scale=0,sill=0)
upper1= list(mean=I,mean1=I,scale=I,sill=I)

fit2 = GeoFit2(data=temp,coordx=coords,corrmodel=corrmodel,X=X,model="Gaussian",
        neighb=5,distance=distance,radius=radius,sensitivity=TRUE,
        optimizer=optimizer, lower=lower1,upper=upper1,
        likelihood="Marginal",type='Pairwise',
         start=start1,fixed=fixed1)
print(fit2)
```

Note that the option *neighb=5* set the compact support of the weight function i.e. $k = 5$ in (10). The object `fit2` include information about the pairwise likelihood estimation:

```
fit2
#################################################################
Maximum  Composite-Likelihood Fitting of Gaussian Random Fields
Setting: Marginal Composite-Likelihood
Model: Gaussian
Type of the likelihood objects: Pairwise
Covariance model: Matern
Optimizer: nlminb
Number of spatial coordinates: 446
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: -11545.30
Estimated parameters:
  mean    mean1    scale     sill
 5.665    1.056   44.565   11.070
#################################################################
```

We now estimate the t random field Y_ν using the function `GeoFit2`. As argued in Bevilacqua et al. (2020), the degrees of freedom parameter ν must be fixed to a positive integer value greater than two. 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. In the second step we fix degrees of freedom by rounding the estimation obtained at the first step. Note that, the function `GeoFit2` works with the inverse of ν, a parametrization suggested in Bevilacqua et al. (2020). As a consequence, the parametric space for the inverse of ν is $(0, 0.5)$. This is the code for the first step estimation:

```
df=4
start1=list(mean=mean,mean1=mean1, scale=scale,df=1/df,sill=sill)
lower2=list(mean=-I,mean1=-I,scale=0,sill=0,df=0)
upper2= list(mean=I,mean1=I,scale=I,sill=I,df=0.5)
fit3 = GeoFit2(data=temp,coordx=coords,corrmodel=corrmodel,
             model="StudentT",neighb=5,distance=distance,X=X,
             radius=radius,sensitivity=TRUE,optimizer=optimizer,
             likelihood="Marginal",type='Pairwise',
             lower=lower2,upper=upper2,start=start1,fixed=fixed1)
```

```
print(fit3)
##################################################################
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: 446
Number of dependent temporal realisations: 1
Type of the random field: univariate
Number of estimated parameters: 5
Type of convergence: Successful
Maximum log-Composite-Likelihood value: -11411.53
Estimated parameters:
     df      mean     mean1     scale      sill
 0.1655   6.1502   1.0253  55.1593   7.4749
##################################################################
```

To guarantee the existence of the t random field we need to round the estimation of ν obtained at first step:

```
DF=as.numeric(round(1/unlist(fit3$param)['df']))
if(DF==2) DF=3
print(DF)
[1] 6
```

Then, we perform the second step estimation keeping fixed the degrees of freedom:

```
fixed=list(nugget=nugget,df=1/DF,smooth=smooth)
start=list(mean=mean,mean1=mean1, scale=scale,sill=sill)
fit4 = GeoFit2(data=temp,coordx=coords,corrmodel=corrmodel,X=X,
             optimizer=optimizer,neighb=5,distance=distance,
             radius=radius, sensitivity=TRUE,lower=lower1,
             likelihood="Marginal",type='Pairwise',
             upper=upper1,start=start,fixed=fixed, model="StudentT")
print(fit4)
```

The object `fit4` include information about the pairwise likelihood estimation:

```
fit4
##################################################################
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: 446
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: -11411.53
Estimated parameters:
  mean    mean1    scale     sill
 6.151    1.025   55.213    7.460
##################################################################
```

It can be appreciated that the t case shows a better maximum log-(Composite) Likelihood value, as expected, since the Gaussian RF is a limit case of the t RF.

Standard error estimation can be performed trough parametric boostrap using the function `GeoVarestbootstrap` that can be computationally demanding in particular for the t case.

```
KK=100
v1=GeoVarestbootstrap(fit2,K=KK,optimizer=optimizer,
                       lower=lower1,upper=upper1,seed=9)#Gaussian
v2=GeoVarestbootstrap(fit4,K=KK,,optimizer=optimizer,
                       lower=lower1,upper=upper1,seed=9)#T
```

The objects `v1` and `v2` contain informations on standard error estimation of the parameters for the Gaussian and t cases and on CLIC and BLIC values used for model selection (Varin and Vidoni, 2005).

```
v1$stderr;v1$claic;v1$clbic
0.65144766 0.04296246 6.00648510 0.86328183
[1] 23207.21
```

```
[1] 23446.26
##
v2$stderr;v2$claic;v2$clbic
     mean       mean1       scale        sill
0.68798019 0.04489521 8.35881351 0.89459516
[1] 22966.5
[1] 23260.58
```

It can be appreciated that both CLIC and BLIC select the t random field.

Checking model assumptions

Given the estimation of the Gaussian and t random fields, the estimated residuals are given by

$$\widehat{G(\boldsymbol{s}_i)} = \frac{g(\boldsymbol{s}_i) - X^\top(\boldsymbol{s})\widehat{\boldsymbol{\beta}}}{(\widehat{\sigma}^2)^{\frac{1}{2}}} \quad i = 1, \dots N \tag{12}$$

and

$$\widehat{Y_\nu(\boldsymbol{s}_i)} = \frac{y(\boldsymbol{s}_i) - X^\top(\boldsymbol{s})\widehat{\boldsymbol{\beta}}}{(\widehat{\sigma}^2)^{\frac{1}{2}}} \quad i = 1, \dots N \tag{13}$$

$\widehat{G(\boldsymbol{s}_i)}$, for $i = 1, \dots N$ can be viewed as a realization of a standard Gaussian random field G^* with marginal distribution $N(0,1)$ and with correlation function $\rho(d_{GC})$. Similarly $\widehat{Y_\nu(\boldsymbol{s}_i)}$ for $i = 1, \dots N$ can be viewed as a realization of a random field Y_ν^* with marginal distribution $t(\nu, 0, 1)$ and with correlation function $\rho_{Y_\nu}(d_{GC})$. The estimated residuals can be computed using the `GeoResiduals` function:

```
res_g=GeoResiduals(fit2); # residuals of Gaussian Random field
res_t=GeoResiduals(fit4); # residuals of t Random field
```

The marginal distribution assumption on the residuals can be graphically checked for instance with a qq-plot (see, Figure (5)) using the function `GeoQQ`:

```
### checking model  residuals assumptions: marginal distribution
GeoQQ(res_g);   #qq-plot residuals of Gaussian Random fields
GeoQQ(res_t);   #qq-plot residuals of t Random fields
```

It can be appreciated that the t case shows a better agreement between the theoretical and estimated quantiles with respect to the Gaussian case. Additionally, the covariance model assumption can be checked comparing the empirical and the estimated semi-variogram of the residuals using the `GeoVariogram` and `GeoCovariogram` functions (see Figure (5)).

```
### semi-variogram residuals of Gaussian Random fields
varionorm=GeoVariogram(data=res_g$data,coordx=coords,radius=radius,
 maxdist=maxdist/5,distance=distance);
GeoCovariogram(res_g,show.vario=TRUE, vario=varionorm,pch=20,ylim=c(0,2.5))

### semi-variogram residuals of t Random fields
variot=GeoVariogram(data=res_t$data,coordx=coords,
maxdist=maxdist/5, distance=distance ,radius=radius);
GeoCovariogram(res_t,show.vario=TRUE, vario=variot,pch=20,ylim=c(0,2.5));
```

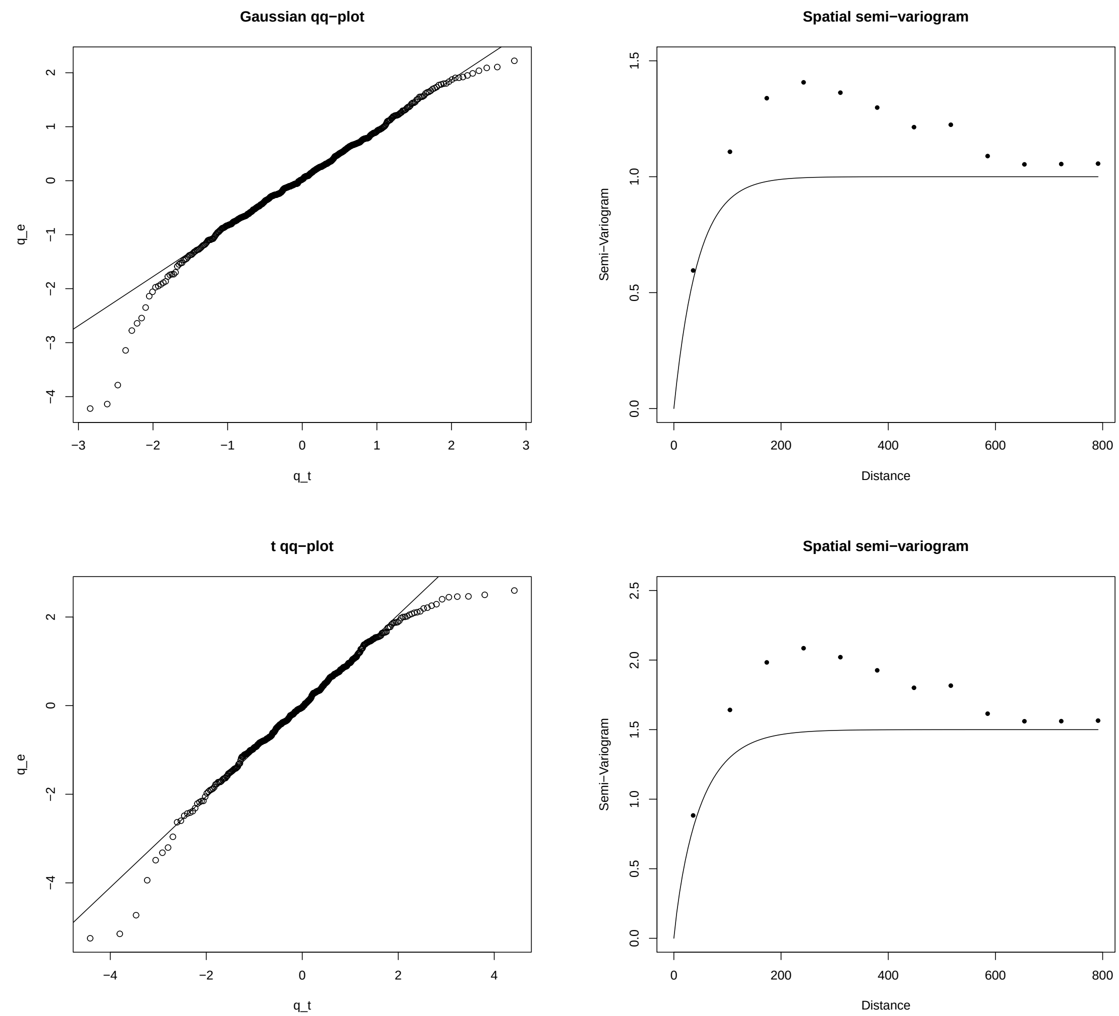

Figure 5: Upper part: qq-plot of the Gaussian residuals and empirical vs estimated semi-variogram of the residuals (from left to right). Bottom part: qq-plot of the t residuals and empirical vs estimated semi-variogram of the residuals (from left to right).

Prediction

The package `GeoModels` allows to perform optimal linear prediction for the Gaussian and t RFs. In the Gaussian case optimal linear prediction is equal to optimal prediction (in the mean squared sense). For a given location $\boldsymbol{s}_0$, the optimal linear prediction of a Gaussian or t RFs is given by:

$$\widehat{L}(\boldsymbol{s}_0) = X(\boldsymbol{s}_0)^T\widehat{\boldsymbol{\beta}} + \boldsymbol{c}^T R^{-1}[\boldsymbol{l} - X\widehat{\boldsymbol{\beta}}], \tag{14}$$

with $\widehat{L}(\boldsymbol{s}_0) = \widehat{G}(\boldsymbol{s}_0)$, $\boldsymbol{l} = \boldsymbol{G}$ or $\widehat{L}(\boldsymbol{s}_0) = \widehat{Y}_\nu(\boldsymbol{s}_0)$, $\boldsymbol{l} = \boldsymbol{Y}_\nu$ for the Gaussian and t cases respectively. In addition:

- $\boldsymbol{c} = (cor(L(\boldsymbol{s}_0), L(\boldsymbol{s}_1)), \ldots, cor(L(\boldsymbol{s}_0), L(\boldsymbol{s}_N)))^T$.
- $R = [\mathrm{cor}(L(\boldsymbol{s}_i), L(\boldsymbol{s}_j)]]_{i,j=1}^N$.

both R and $\boldsymbol{c}$ are computed by using the estimated correlation functions $\rho(d_{GC})$ and $\rho_{Y_\nu}(d_{GC})$ for the Gaussian and t case respectively. Moreover the associated mean square error (MSE) is given by:

$$MSE(\widehat{L}(\boldsymbol{s}_0)) = \widehat{Var(L(\boldsymbol{s}}))(1 - \boldsymbol{c}^T R^{-1}\boldsymbol{c}). \tag{15}$$

where $\widehat{Var(L(\boldsymbol{s}}))$ is given by $\widehat{Var(G(\boldsymbol{s}})) = \widehat{\sigma}^2$ and $\widehat{Var(Y_\nu(\boldsymbol{s}}))= \widehat{\sigma}^2\nu/(\nu-2)$ for the Gaussian and t RF respectively.

Prediction can be performed with the `GeoKrig` function. For instance, computing the prediction of the residual at point with lon-lat coordinates $(135, -25)^T$ can be performed with the following code:

```
coords_to_pred=matrix(c(135,-25),ncol=2)
prstudent<-GeoKrig(data=res_t$data, coordx=coords,loc=coords_to_pred,
corrmodel=corrmodel,distance=distance,radius=radius,mse=TRUE,
model="StudentT",param= append(res_t$param,res_t$fixed))
```

Prediction and associated estimated MSE (15) of the residual can be obtained from the object `prstudent`:

```
pr_student2$pred
[1] 0.04237531
pr_student2$mse
[1]  1.497777
```

We can evaluate the predictive performances of the Gaussian and t RFs using cross validation, with the function `GeoCV`. The procedure can be computationally demanding in particular for large dataset and/or when the estimation/prediction procedures involve special functions as in the case of t random field.

```
KK=100
d=GeoCV(fit2,K=KK,n.fold=0.2,seed=9)
[1] ‘Cross-validation  kriging can be time consuming ...‘
[1] ‘Starting iteration from 1 to 100  ...‘
e=GeoCV(fit4,K=KK,n.fold=0.2,seed=9)
[1] ‘Cross-validation  kriging can be time consuming ...
[1] ‘Starting iteration from 1 to 100  ...
```

The function basically randomly choose 80% of the spatial locations for estimation and use the remaining 20% as data for the predictions, where the (optimal linear) predictions are internally obtained using `GeoKrig` function. Then some prediction scores as RMSE and MAE (Gneiting and Raftery, 2007) are constructed by comparing the predictions with with the (known) values. This is iterated 100 times. For instance we can compare the prediction performance of the Gaussian and t random fields using the empirical mean of the 100 RMSEs and MAEs

```
> mean(e$rmse);mean(e$mae); #  gaussian
[1]2.815366
[1]2.167679
mean(d$rmse);mean(d$mae); #  T
[1] 2.787085
[1] 2.150952
```

It can be appreciated that the estimated t RF perform better from prediction viewpoint using the RMSE and MAE criterion, even if the optimal linear prediction is not optimal in the t case.

Finally, a kriging map with associated MSE can be obtained using the `GeoKrig` function. For the given location sites, we first need to specify the border of the region and then to construct a fine grid inside the border. The following code perform this task:

```
coord <- maps:::map.poly("worldHires", "Australia", exact=FALSE,
                         xlim=c(110,160),ylim=c(-45,-5),
                         boundary=FALSE,
```

```
                              interior=TRUE, fill=FALSE, as.polygon=TRUE)
coord.sp <- map2SpatialPolygons(coord,coord$names)

long1=110;long2=154
lat1=-39;lat2=-12
lat_seq=seq(lat1,lat2,0.25)
lon_seq=seq(long1,long2,0.25)
coords_tot=as.matrix(expand.grid(lon_seq,lat_seq))
gr.in <- locations.inside(coords_tot, coord.sp)
plot(gr.in)
```

Then (optimal) linear prediction (14) and associated MSE (15) can be computed (using the estimated parameters) for the Gaussian and t cases, with the following code:

```
pr_gaussian<-GeoKrig(data=res_g$data, coordx=coords,loc=gr.in,
  corrmodel=corrmodel,distance=distance,radius=radius,
  mse=TRUE,model="Gaussian",param= as.list(c(res_g$param,res_g$fixed)))

pr_student2<-GeoKrig(data=res_t$data, coordx=coords,loc=gr.in,
  corrmodel=corrmodel,distance=distance,radius=radius,mse=TRUE,
  model="StudentT",param= as.list(c(res_t$param,res_t$fixed)))
```

Finally a kriging map with associated mean square error (Figure 6) can be obtained with the following code:

```
quilt.plot(gr.in,pr_gaussian$pred)
oz(states=FALSE,add=T, lwd=2)
quilt.plot(gr.in,pr_gaussian$mse)
oz(states=FALSE,add=T, lwd=2)
quilt.plot(gr.in,pr_student2$pred)
oz(states=FALSE,add=T, lwd=2)
quilt.plot(gr.in,pr_student2$mse)
oz(states=FALSE,add=T, lwd=2)
```

References

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

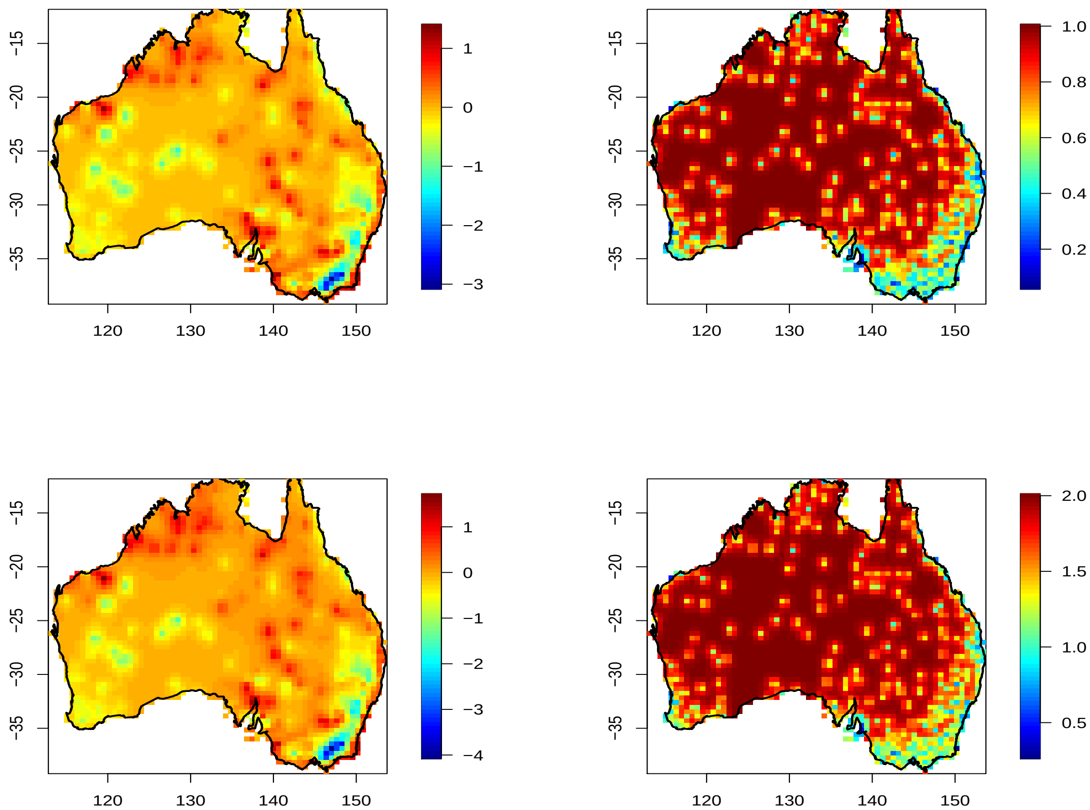

Figure 6: Kriging map and mean squared error map for the residuals of the estimated Gaussian (first row) and t (second row) random fields.

Bevilacqua, M., C. Caamaño-Carrillo, R. B. A. Valle, and V. Morales-Oñate (2020). Non-gaussian geostatistical modeling using (skew) t processes. *Scand J Statist*, 1–34.

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., 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.

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

Gneiting, T. and A. E. Raftery (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association 102*, 359–378.

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.

Kilibarda, M., T. Hengl, G. B. M. Heuvelink, B. Gräler, E. Pebesma, M. Perčec Tadić, and B. Bajat (2014). Spatio-temporal interpolation of daily temperatures for global land areas at 1km resolution. *Journal of Geophysical Research: Atmospheres 119*(5), 2294–2313.

Varin, C. and P. Vidoni (2005). A note on composite likelihood inference and model selection. *Biometrika 52*, 519–528.