Revista Colombiana de Estadística
Diciembre 2012, volumen 35, no. 3, pp. 385 a 407

geofd: An R Package for Function-Valued Geostatistical Prediction

geofd: un paquete R para predicción geoestadística de datos funcionales

Ramón Giraldo[1,a], Jorge Mateu[2,b], Pedro Delicado[3,c]

[1]Department of Statistics, Sciences Faculty, Universidad Nacional de Colombia, Bogotá, Colombia

[2]Department of Mathematics, Universitat Jaume I, Castellón, Spain

[3]Department of Statistics and Operations Research, Universitat Politècnica de Catalunya, Barcelona, Spain

Abstract

Spatially correlated curves are present in a wide range of applied disciplines. In this paper we describe the R package geofd which implements ordinary kriging prediction for this type of data. Initially the curves are pre-processed by fitting a Fourier or B-splines basis functions. After that the spatial dependence among curves is estimated by means of the trace-variogram function. Finally the parameters for performing prediction by ordinary kriging at unsampled locations are by estimated solving a linear system based estimated trace-variogram. We illustrate the software analyzing real and simulated data.

***Key words*:** Functional data, Smoothing, Spatial data, Variogram.

Resumen

Curvas espacialmente correlacionadas están presentes en un amplio rango de disciplinas aplicadas. En este trabajo se describe el paquete R geofd que implementa predicción por kriging ordinario para este tipo de datos. Inicialmente las curvas son suavizadas usando bases de funciones de Fourier o B-splines. Posteriormente la dependencia espacial entre las curvas es estimada por la función traza-variograma. Finalmente los parámetros del predictor kriging ordinario son estimados resolviendo un sistema de ecuaciones basado en la estimación de la función traza-variograma. Se ilustra el paquete analizando datos reales y simulados.

***Palabras clave*:** datos funcionales, datos espaciales, suavizado, variograma.

[a]Associate professor. E-mail: rgiraldoh@unal.edu.co
[b]Professor. E-mail: mateu@mat.uji.es
[c]Associate professor. E-mail: pedro.delicado@upc.edu

1. Introduction and Overview

The number of problems and the range of disciplines where the data are functions has recently increased. This data may be generated by a large number of measurements (over time, for instance), or by automatic recordings of a quantity of interest. Since the beginning of the nineties, functional data analysis (FDA) has been used to describe, analyze and model this kind of data. Functional versions for a wide range of statistical tools (ranging from exploratory and descriptive data analysis to linear models and multivariate techniques) have been recently developed (see an overview in Ramsay & Silverman 2005). Standard statistical techniques for FDA such as functional regression (Malfait & Ramsay 2003) or functional ANOVA (Cuevas, Febrero & Fraiman 2004) assume independence among functions. However, in several disciplines of the applied sciences there exists an increasing interest in modeling correlated functional data: This is the case when functions are observed over a discrete set of time points (temporally correlated functional data) or when these functions are observed in different sites of a region (spatially correlated functional data). For this reason, some statistical methods for modeling correlated variables, such as time series (Box & Jenkins 1976) or spatial data analysis (Cressie 1993), have been adapted to the functional context. For spatially correlated functional data, Yamanishi & Tanaka (2003) developed a regression model that enables to model the relationship among variables over time and space. Baladandayuthapani, Mallick, Hong, Lupton, Turner & Caroll (2008) showed an alternative for analyzing an experimental design with a spatially correlated functional response. For this type of modeling an associate software in MATLAB (MATLAB 2010) is available at http://odin.mdacc.tmc.edu/∼vbaladan. Staicu, Crainiceanu & Carroll (2010) propose principal component-based methods for the analysis of hierarchical functional data when the functions at the lowest level of the hierarchy are correlated. A software programme accompanying this methodology is available at http://www4.stat.ncsu.edu/∼staicu. Delicado, Giraldo, Comas & Mateu (2010) give a review of some recent contributions in the literature on spatial functional data. In the particular case of data with spatial continuity (geostatistical data) several kriging and cokriging predictors (Cressie 1993) have been proposed for performing spatial prediction of functional data. In these approaches a smoothing step, usually achieved by means of Fourier or B-splines basis functions, is initially carried out. Then a method to establish the spatial dependence between functions is proposed and finally a predictor for carrying out spatial prediction of a curve on a unvisited location is considered. Giraldo, Delicado & Mateu (2011) propose a classical ordinary kriging predictor, but considering curves instead of one-dimensional data; that is, each curve is weighted by a scalar parameter. They called this method "ordinary kriging for function-valued spatial data" (OKFD). This predictor was initially considered by Goulard & Voltz (1993). On the other hand; Giraldo, Delicado & Mateu (2010) solve the problem of spatial prediction of functional data by weighting each observed curve by a functional parameter. Spatial prediction of functional data based on cokriging methods are given in Giraldo (2009) and Nerini, Monestiez & Manté (2010). All of abovementioned approaches are important from a theoretical and applied perspective.

A comparison of these methods based on real data suggests that all of them are equally useful (Giraldo 2009). However, from a computational point of view the approach based on OKFD is the simplest because the parameters to estimate are scalars. In other cases the parameters are functions themselves and in addition it is necessary to estimate a linear model of coregionalization (Wackernagel 1995) for modeling the spatial dependence among curves, which could be restrictive when the number of basis functions used for smoothing the data set is large. For this reason the current version of the package geofd implemented within the statistical environment R (R Development Core Team 2011) only contains functions for doing spatial prediction of functional data by OKFD. However, the package will be progressively updated including new R functions.

It is important to clarify that the library geofd allows carrying out spatial prediction of functional data (we can predict a whole curve). This software cannot be used for doing spatio-temporal prediction. There is existing software that analyzes and models space-time data by considering a space-time covariance model and using to make this model predictions. There is no existing software for functional spatial prediction except the one we present in this paper. We believe there is no reason for confusion and the context gives us the necessary information to use existing space-time software or our software.

The package geofd has been designed mainly to support teaching material and to carry out data analysis and simulation studies for scientific publications. Working in geofd with large data sets can be a problem because R has limited memory to deal with such a large object. A solution can be use R packages for big data support such as bigmemory (http://www.bigmemory.org) or ff (http://ff.r-forge.r-project.org/).

This work is organized as follows: Section 2 gives a brief overview of spatial prediction by means of OKFD method, Section 3 describes the use of the package geofd based on the analysis of real and simulated data and conclusions are given in Section 4.

2. Ordinary Kriging for Functional Data

Ferraty & Vieu (2006) define a *functional variable* as a random variable X taking values in an infinite dimensional space (or functional space). *Functional data* is an observation x of X. A *functional data set* $x_1, \ldots, x_n$ is the observation of n functional variables $X_1, \ldots, X_n$ distributed as X. Let $T = [a, b] \subseteq \mathbb{R}$. We work with functional data that are elements of

$$L_2(T) = \{X : T \to \mathbb{R}, \text{ such that } \int_T X(t)^2 dt < \infty\}$$

Note that $L_2(T)$ with the inner product $\langle x, y\rangle = \int_T x(t)y(t)dt$ defines an Euclidean space.

Following Delicado et al. (2010) we define a functional random process as $\{X_s(t) : s \in D \subseteq \mathbb{R}^d, t \in T \subseteq \mathbb{R}\}$, usually $d = 2$, such that $X_s(t)$ is a functional variable for any $s \in D$. Let $s_1, \ldots, s_n$ be arbitrary points in D and assume

that we can observe a realization of the functional random process $X_s(t)$ at these n sites, $x_{s_1}(t), \ldots, x_{s_n}(t)$. OKFD is a geostatistical technique for predicting $X_{s_0}(t)$, the functional random process at s_0, where s_0 is a unsampled location.

It is usually assumed that the functional random process is second-order stationary and isotropic, that is, the mean and variance functions are constant and the covariance depends only on the distance between sampling points (however, the methodology could also be developed without assuming these conditions). Formally, we assume that

1. $E(X_s(t)) = m(t)$ and $V(X_s(t)) = \sigma^2(t)$ for all $s \in D$ and all $t \in T$.
2. $COV(X_{s_i}(t), X_{s_j}(t)) = C(\|s_i - s_j\|)(t) = C_{ij}(h,t), s_i, s_j \in D, t \in T$, where $h = \|s_i - s_j\|$.
3. $\frac{1}{2}V(X_{s_i}(t) - X_{s_j}(t)) = \gamma(\|s_i - s_j\|)(t) = \gamma(h,t), s_i, s_j \in D, t \in T$, where $h = \|s_i - s_j\|$.

These assumptions imply that $V(X_{s_i}(t) - X_{s_j}(t)) = E(X_{s_i}(t) - X_{s_j}(t))^2$ and $\gamma\|s_i - s_i\|(t) = \sigma^2(t) - C(\|s_i - s_j\|)(t)$.

The OKFD predictor is defined as (Giraldo et al. 2011)

$$\widehat{X}_{s_0}(t) = \sum_{i=1}^{n} \lambda_i X_{s_i}(t), \;\; \lambda_1, \ldots, \lambda_n \in \mathbb{R} \tag{1}$$

The predictor (1) has the same expression as the classical ordinary kriging predictor (Cressie 1993), but considering curves instead of variables. The predicted curve is a linear combination of observed curves. Our approach considers the whole curve as a single entity, that is, we assume that each measured curve is a complete datum. The kriging coefficients or weights λ in Equation (1) give the influence of the curves surrounding the unsampled location where we want to perform our prediction. Curves from those locations closer to the prediction point will naturally have greater influence than others more far apart. These weights are estimated in such a way that the predictor (1) is the best linear unbiased predictor (BLUP). We assume that each observed function can be expressed in terms of K basis functions, $B_1(t), \ldots, B_K(t)$, by

$$x_{s_i}(t) = \sum_{l=1}^{K} a_{il} B_l(t) = \boldsymbol{a}_i^T \boldsymbol{B}(t), \; i = 1, \ldots, n \tag{2}$$

where $\boldsymbol{a}_i = (a_{i1}, \ldots, a_{iK})$, $\boldsymbol{B}(t) = (B_1(t), \ldots, B_K(t))$

In practice, these expressions are truncated versions of Fourier series (for periodic functions, as it is the case for Canadian temperatures) or B-splines expansions. Wavelets basis can also be considered (Giraldo 2009).

To find the BLUP, we consider first the unbiasedness. From the constant mean condition above, we require that $\sum_{i=1}^{n} \lambda_i = 1$. In a classical geostatistical setting we assume that the observations are realizations of a random field

$\{X_s : s \in D, D \in \mathbb{R}^d\}$. The kriging predictor is defined as $\widehat{X}_{s_0} = \sum_{i=1}^{n} \lambda_i X_{s_i}$, and the BLUP is obtained by minimizing

$$\sigma^2_{s_0} = V(\widehat{X}_{s_0} - X_{s_0})$$

subject to $\sum_{i=1}^{n} \lambda_i = 1$. On the other hand in multivariable geostatistics (Myers 1982, Ver Hoef & Cressie 1993, Wackernagel 1995) the data consist of $\{\mathbf{X}_{s_1}, \ldots, \mathbf{X}_{s_n}\}$, that is, we have observations of a spatial vector-valued process $\{\mathbf{X}_s : s \in D\}$, where $\mathbf{X}_s = (X_s(1), \ldots, X_s(m))$ and $D \in \mathbb{R}^d$. In this context $V(\widehat{\boldsymbol{X}}_{s_0} - \boldsymbol{X}_{s_0})$ is a matrix, and the BLUP of m variables at an unsampled location s_0 can be obtained by minimizing

$$\sigma^2_{s_0} = \sum_{i=1}^{m} V\left(\widehat{X}_{s_0}(i) - X_{s_0}(i)\right)$$

subject to constraints that guarantee unbiasedness conditions, that is, minimizing the trace of the mean-squared prediction error matrix subject to some restrictions given by the unbiasedness condition (Myers 1982). Extending the criterion given in Myers (1982) to the functional context by replacing the summation by an integral, the n parameters in Equation (1) are obtained by solving the following constrained optimization problem (Giraldo et al. 2011)

$$\min_{\lambda_1,\ldots,\lambda_n} \int_T V(\widehat{X}_{s_0}(t) - X_{s_0}(t))dt, \text{ s.t.} \sum_{i=1}^{n} \lambda_i = 1 \tag{3}$$

which after some algebraic manipulation can be written as

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j \int_T C_{ij}(h,t)dt + \int_T \sigma^2(t)dt - 2\sum_{i=1}^{n} \int_T C_{i0}(h,t)dt + 2\mu(\sum_{i=1}^{n} \lambda_i - 1) \tag{4}$$

where μ is the Lagrange multiplier used to take into account the unbiasedness restriction. Minimizing (4) with respect to $\lambda_1, \ldots, \lambda_n$ and μ, we find the following linear system which enables to estimate the parameters

$$\begin{pmatrix} \int_T \gamma\|s_1 - s_1\|(t)dt & \cdots & \int_T \gamma\|s_1 - s_n\|(t)dt & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \int_T \gamma\|s_n - s_1\|(t)dt & \cdots & \int_T \gamma\|s_n - s_n\|(t)dt & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ -\mu \end{pmatrix}$$
$$= \begin{pmatrix} \int_T \gamma\|s_0 - s_1\|(t)dt \\ \vdots \\ \int_T \gamma\|s_0 - s_n\|(t)dt \\ 1 \end{pmatrix} \tag{5}$$

The function $\gamma(h) = \int_T \gamma\|s_i - s_j\|(t)dt$, is called the trace-variogram. In order to solve the system in (5), an estimator of the trace-variogram is needed. Given that we are assuming that $X_s(t)$ has a constant mean function $m(t)$ over D, $V(X_{s_i}(t) - X_{s_j}(t)) = E[(X_{s_i}(t) - X_{s_j}(t))^2]$. Note that, using Fubini's theorem

$$\gamma(h) = \frac{1}{2} E\left[\int_T (X_{s_i}(t) - X_{s_j}(t))^2 dt\right], \text{ for } s_i, s_j \in D \text{ with } h = \|s_i - s_j\| \tag{6}$$

Then an adaptation of the classical method-of-moments (MoM) for this expected value, gives the following estimator

$$\widehat{\gamma}(h) = \frac{1}{2|N(h)|} \sum_{i,j \in N(h)} \int_T (X_{s_i}(t) - X_{s_j}(t))^2 dt \tag{7}$$

where $N(h) = \{(s_i, s_j) : \|s_i - s_j\| = h\}$, and $|N(h)|$ is the number of distinct elements in $N(h)$. For irregularly spaced data there are generally not enough observations separated by exactly a distance h. Then $N(h)$ is modified to $\{(s_i, s_j) : \|s_i - s_j\| \in (h - \varepsilon, h + \varepsilon)\}$, with $\varepsilon > 0$ being a small value.

Once we have estimated the trace-variogram for a sequence of K values h_k, a parametric model $\gamma(h; \theta)$ such as spherical, Gaussian, exponential or Matérn (Ribeiro & Diggle 2001) must be fitted.

The prediction trace-variance of the functional ordinary kriging based on the trace-variogram is given by

$$\sigma^2_{s_0} = \int_T V(\widehat{X}_{s_0}(t) - X_{s_0}(t))dt = \sum_{i=1}^{n} \lambda_i \int_T \gamma \|s_i - s_0\|(t)dt - \mu \tag{8}$$

This parameter should be considered as a global uncertainty measure, in the sense that it is an integrated version of the classical pointwise prediction variance of ordinary kriging. For this reason its estimation cannot be used to obtain a confidence interval for the predicted curve. There is not, to the best of our knowledge, a method which allows us to do spatial prediction of functional data with an estimation of a prediction variance curve. We must take into account that we predict a whole curve and is not possible with this methodology to get point-wise confidence intervals, as we can obtain by using space or space-time models. It is clear that spatial-functional data and spatial temporal models have a common link in the sense that we have evolution of a spatial process through time or through any other characteristic. But at the same time there is an important difference. Spatial temporal models consider the evolution of a spatial process through time and models the interdependency of space and time. In this case we have $X(s, t)$ a single variable and we want to predict a variable at an unsampled location. In the spatial-functional case $X_s(t)$ is itself a function and thus we aim at predicting a function.

3. Illustration

Table 1 summarizes the functions of the package geofd. To illustrate its use we analyze real and simulated data. Initially in Sections 3.1 and 3.2 we apply the methodology to temperature measurements recorded at 35 weather stations located in the Canadian Maritime Provinces (Figure 1, left panel). Then the results with a simulated data set are shown in Section 3.3

The Maritime Provinces cover a region of Canada consisting of three provinces: Nova Scotia (NS), New Brunswick (NB), and Prince Edward Island (PEI). In particular, we analyze information of daily mean temperatures averaged over the

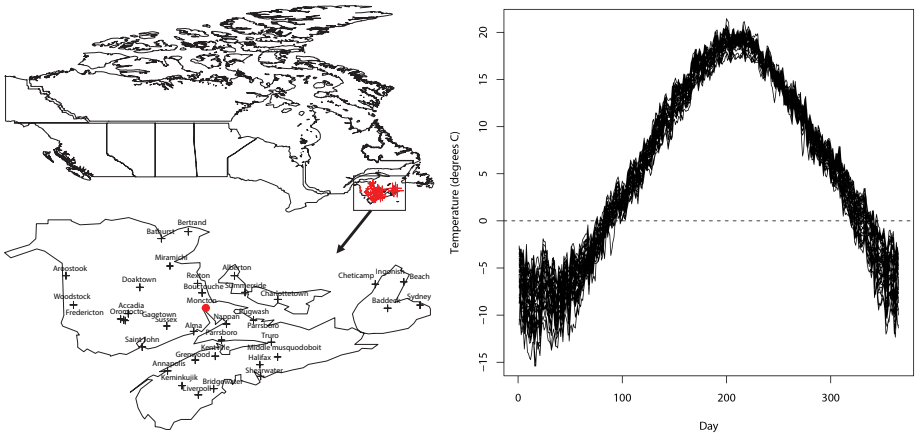

FIGURE 1: Averages (over 30 years) of daily temperature curves (right panel) observed at 35 weather stations of the Canadian Maritime provinces (left panel).

TABLE 1: Summary of the geofd functions.

Function	Description
`fit.tracevariog`	Fits a parametric model to the trace-variogram
`.geofd.viewer`	Graphical interface to plot multiple predictions
`l2.norm`	Calculates the L_2 norm between all pairs of curves
`maritimes.data`	Temperature values at 35 weather stations of Canada
`maritimes.avg`	Average temperature at Moncton station
`okfd`	Ordinary kriging for function-value data
`okfd.cv`	Cross-validation analysis for ordinary kriging for function-value data
`plot.geofd`	Plot the trace-variogram function and some adjusted models
`trace.variog`	Calculates the trace-variogram function

years 1960 to 1994 (February 29th combined with February 28th) (Figure 1, right panel). The data for each station were obtained from the Meteorological Service of Canada (http://www.climate.weatheroffice.ec.gc.ca/climateData/). Our package makes use of the R libraries fda (Ramsay, Hooker & Graves 2009) for smoothing data (by Fourier or B-splines basis) and geoR (Ribeiro & Diggle 2001) for fitting a variogram model to the estimated trace-variogram function. The temperature data set considered (Figure 1, right panel) is periodic and consequently a Fourier basis function is the most appropriate choice for smoothing it (Ramsay & Silverman 2005). However for illustrative purposes we also use a B-spline basis function. We can make a prediction at only one site or at multiple locations. Both alternatives are considered in the examples (Figure 2). In Section 3.1 we smooth the temperature data using a B-splines basis and, make a prediction at an unvisited location (left panel, Figure 2). In Section 3.2 we smooth the data using a Fourier basis and predict the temperature curves at ten randomly chosen sites (right panel, Figure 2).

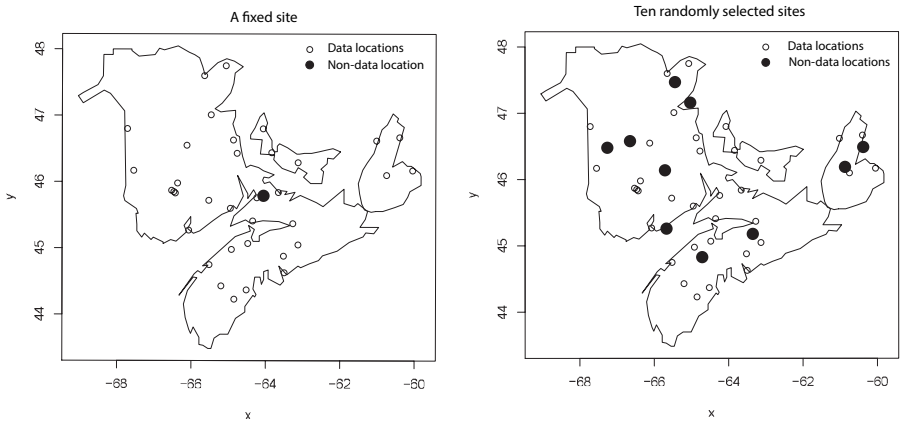

FIGURE 2: Prediction sites. A fixed site considered in the first example (left panel) and ten randomly selected sites considered in the second one (right panel).

3.1. Using a B-splines Basis

The following code illustrates how to use the library geofd for predicting a temperature curve at an unsampled location when the data are smoothed by using a B-splines basis. Initially we read and plot the data set (Figure 1, right panel), plot the coordinates of visited sites and choose a site for carrying out a prediction (Figure 2, left panel). The R code is the following.

```
R> library (geofd)
R> data(maritimes)
```

The `library(geofd)` command loads the package geofd (and other dependent packages) into the R computing environment. The `data(maritimes)` command loads the maritimes data set containing 35 temperature curves obtained at the same number of weather stations of the maritime provinces of Canada. The first five temperature values for four weather stations are

```
R> head(maritimes.data[,1:4], n=5)
```

```
     Fredericton      Halifax Sydney    Miramichi
[1,]        -7.9    -4.4      -3.8     -8.60
[2,]        -7.5    -4.2      -3.5     -8.32
[3,]        -9.3    -5.3      -4.6     -9.87
[4,]        -8.7    -5.4      -5.0     -9.55
[5,]        -9.1    -5.6      -4.1     -9.58
```

The next five lines of commands allow to plot the data and the coordinates.

```
R> matplot(maritimes.data,type="l",xlab="Day",ylab="degress C")
R> abline(h=0, lty=2)
```

```
R> plot(maritimes.coords)
R> coord.cero <- matrix(c(-64.06, 45.79),nrow=1,ncol=2)
R> points(coord.cero, col=2, lwd=3)
```

The main function of geofd is `okfd` (Table 1). This function allows to carry out predictions by ordinary kriging for function-valued data by considering a Fourier or a B-splines basis as methods for smoothing the observed data set. This covers from the smoothing step and trace-variogram estimation to data prediction. Although the estimation of the trace-variogram can be obtained by directly using the function `okfd`, it is also possible to estimate it in a sequential way by using the functions `l2.norm`, `trace.vari` and `fit.tracevariog`, respectively (Table 1). Now we give an illustration in this sense. In this example the data set is smoothed by using a B-splines basis with 65 functions without penalization (Figure 3, left panel). The number of basis functions was chosen by cross-validation (Delicado et al. 2010). We initially define the parameters for smoothing the data. We use here the fda library. An overview of the smoothing functional data by means of B-splines basis using the library fda library can be found in (Ramsay, Wickham, Graves & Hooker 2010). The following code illustrates how to run this process with the maritime data set.

```
R> n<-dim(maritimes.data)[1]
R> argvals<-seq(1,n, by=1)
R> s<-35
R> rangeval <- range(argvals)
R> norder   <- 4
R> nbasis   <- 65
R> bspl.basis <- create.bspline.basis(rangeval, nbasis, norder)
R> lambda <-0
R> datafdPar <- fdPar(bspl.basis, Lfdobj=2, lambda)
R> smfd <- smooth.basis(argvals,maritimes.data,datafdPar)
R> datafd <- smfd$fd
R> plot(datafd, lty=1, xlab="Day", ylab="Temperature (degrees C)")
```

The smoothed curves are shown in the left panel of Figure 3. Once wi have smoothed the data, we can use the functions above for estimating the trace-variogram. First we have to calculate the L_2 norm between the smoothed curves using the function `l2.norm`. The arguments for this function are the number `s` of sites where curves are observed, `datafd` a functional data object representing a smoothed data set and `M` a symmetric matrix of order equal to the number of basis functions defined by the B-splines basis object, where each element is the inner product of two basis functions after applying the derivative or linear differential operator defined by Lfdobj (Ramsay et al. 2010).

```
R> M <- bsplinepen(bspl.basis,Lfdobj=0)
R> L2norm <- l2.norm(s, datafd, M)
```

In the above commands the results are assigned to the variable `L2norm`. This one stores a matrix whose values correspond to the L_2 norm between each pair

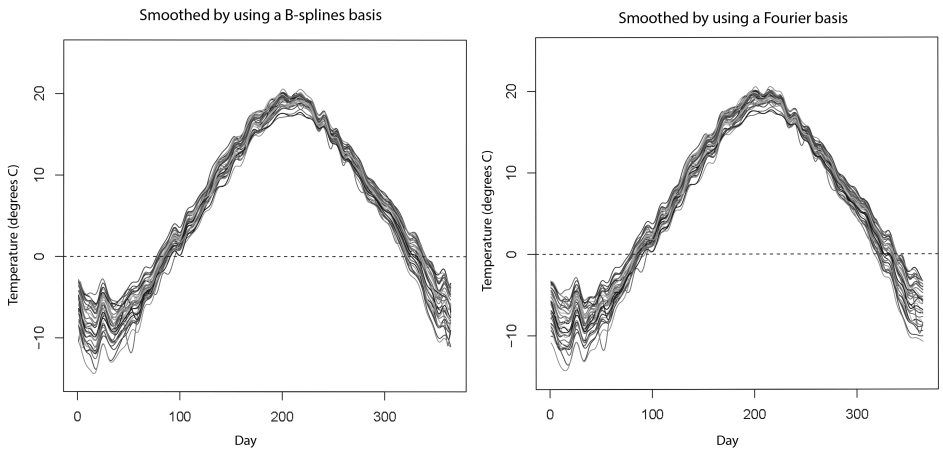

FIGURE 3: Smoothed data of daily temperature by using a B-splines basis (left panel) and a Fourier basis (right panel) with 65 functions.

of functional data into the data set. This matrix is then passed to the function `trace.variog` for estimating the trace-variogram function. The output can be returned as a trace-variogram "cloud" or as a binned trace-variogram (see Equation 7). The following code shows how this function can be used in combination with `fit.tracevariog` for fitting a model to the trace-variogram function obtained with the maritime data set. The main arguments of the function `trace.variog` are `coords` the geographical coordinates in decimal degrees where data where recorded, L_2 `norm` a matrix whose values are the L_2 norm between all pair of smoothed functions (an output from the function `l2.norm`), `bin` which is a logical argument indicating whether the output is a binned variogram, `maxdist` a numerical value defining the maximum distance for calculating the trace-variogram. Other arguments such as `uvec`, `breaks` and `nugget.tolerance` are defined as in the function `variog` of the package geoR. In order to fit a theoretical model (exponential, Gaussian, spherical or Matern) to the estimated trace-variogram we can use the function `fit.tracevariog`. This function makes use of the function `variofit` of geoR. The arguments of these functions are the estimations of the trace-variogram function (an output of the function `trace.variog`), `model` a list with the models that we want to fit, and some initial values for the parameters in these models. The command lines below show the use of these functions.

```
R> dista=max(dist(maritimes.coords))*0.9
R> tracev=trace.variog(maritimes.coords, L2norm, bin=FALSE,
+  max.dist=dista,uvec="default",breaks="default",nugget.tolerance)
R> models=fit.tracevariog(tracev, models=c("spherical","exponential",
+  "gaussian","matern"),sigma2.0=2000, phi.0=4, fix.nugget=FALSE,
+  nugget=0, fix.kappa=TRUE, kappa=1, max.dist.variogram=dista)
```

The variable `tracev` above stores the output of the function `trace.variog` which is used posteriorly in the function `plot.geofd` for plotting the trace-variogram

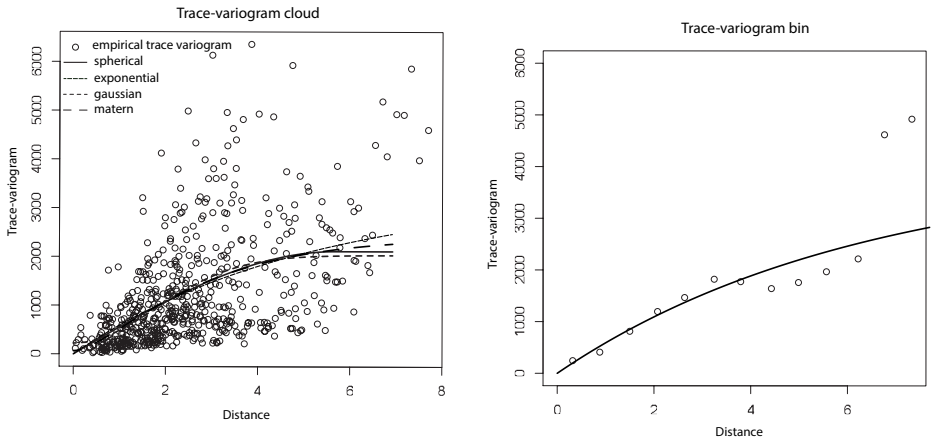

FIGURE 4: Estimated trace-variogram "cloud" and four fitted models (left panel). Estimated trace-variogram "bin" and the best fitted model (right panel).

"cloud". On the other hand the variable `model` stores the results obtained with the function `fit.tracevariog`. The use of the function `plot.geofd` in combination with the command `lines (models$fitted)` produces the plot shown in Figure 4 (left panel), this is, the estimated trace-variogram "cloud" and the four fitted models (exponential, Gaussian, spherical and Matern).

```
R> plot(tracev, xlab="Distancia", ylab="Trace-Variogram")
R> lines(models$fitted[[1]], lwd=2)
R> lines(models$fitted[[2]], lwd=2, col=4)
R> lines(models$fitted[[3]], lwd=2, col=7)
R> lines(models$fitted[[4]], lwd=2, col=6)
R> legend("topleft",   c("empirical trace variogram", "spherical",
+  "exponential", "gaussian",   "matern"), lty=c(-1,1,1,1,1),
+  col=c(1,1,4,7,6), pch=c(1,-1,-1,-1,-1))
```

In Figure 4 (right panel) the estimated trace-variogram "bin" and the best fitted model are shown. This plot is obtained by using the code below. In this case we use the option `bin=TRUE` in the function `trace.variog`, and the command line `lines(models$fitted[[2]], lwd=2, col=4)` to plot the exponential model.

```
R> tracevbin=trace.variog(maritimes.coords, L2norm, bin=TRUE,
+  max.dist=dista)
R> plot(tracevbin$u, tracevbin$v, ylim=c(0,3000), xlim=c(0, 7),
+  xlab="Distance", ylab="Trace-Variogram")
R> lines(models$fitted[[2]], lwd=2, col=4)
```

The numerical results of the function `fit.tracevariog` are stored in the object `models`. This list contains the estimations of the parameters (τ^2, σ^2, and ϕ) for each trace-variogram model and the minimized sum of squared errors (see

`variofit` from geoR). According to the results below we can observe that the best model (least sum of squared errors) is the exponential model.

```
R>models

[[1]]
variofit: model parameters estimated by OLS (ordinary least squares):
covariance model is: spherical
fixed value for tausq =  0
parameter estimates:
  sigmasq       phi
3999.9950   12.0886
Practical Range with cor=0.05 for asymptotic range: 12.08865
variofit: minimised sum of squares = 529334304
[[2]]
variofit: model parameters estimated by OLS (ordinary least squares):
covariance model is: exponential
fixed value for tausq =  0
parameter estimates:
  sigmasq       phi
4000.0003    6.2689
Practical Range with cor=0.05 for asymptotic range: 18.77982
variofit: minimised sum of squares = 524840646
[[3]]
variofit: model parameters estimated by OLS (ordinary least squares):
covariance model is: gaussian
fixed value for tausq =  0
parameter estimates:
  sigmasq       phi
2092.8256    2.2886
Practical Range with cor=0.05 for asymptotic range: 3.961147
variofit: minimised sum of squares = 541151209
fitted[[4]]
variofit: model parameters estimated by OLS (ordinary least squares):
covariance model is: Matern with fixed kappa = 1
fixed value for tausq =  0
parameter estimates:
  sigmasq       phi
2693.1643    1.9739
Practical Range with cor=0.05 for asymptotic range: 7.892865
variofit: minimised sum of squares = 529431348
```

Once fitted, the best trace-variogram model we can use the `okfd` function for performing spatial prediction at an unvisited location. The arguments of this function are `new.coords` an $n \times 2$ matrix containing the coordinates of the new n prediction sites, `coords` an $s \times 2$ matrix containing the coordinates of the s sites where functional data were recorded, `data` an $m \times s$ matrix with values

for the observed functions, `smooth.type` a string with the name of smoothing method to be used (B-splines or Fourier), `nbasis` a numeric value defining the number of basis functions used to smooth the discrete data set recorded at each site, `argvals` a vector containing argument values associated with the values to be smoothed, `lambda` (optional) a penalization parameter for smoothing the observed functions, and `cov.model` a string with the name of the correlation function (see `variofit` from geoR). Other additional arguments are `fix.nugget`, `nugget value`, `fix.kappa`, `kappa` (related to the parameters of the correlation model), and `max.dist.variogram` a numerical value defining the maximum distance considered when fitting the variogram model. The code below allows to predict a temperature curve at the Moncton weather station (see Figure 1).

```
R> okfd.res<-okfd(new.coords=coord.cero, coords=maritimes.coords,
+  cov.model="exponential",  data=maritimes.data, nbasis=65,
+  argvals=argvals, fix.nugget=TRUE)
R> plot(okfd.res$datafd, lty=1,col=8, xlab="Day",
+  ylab="Temperature (degrees C)",
+  main="Prediction at Moncton")
R> lines(okfd.res$argvals, okfd.res$krig.new.data, col=1, lwd=2,
+  type="l", lty=1, main="Predictions", xlab="Day",
+  ylab="Temperature (Degrees C)")
R> lines(maritimes.avg,  type="p", pch=20,cex=0.5, col=2, lwd=1)
```

A graphical comparison between real data (see `maritimes.avg` in Table 1) and the predicted curve (Figure 5) allows to conclude that the method OKFD has a good performance with this data set.

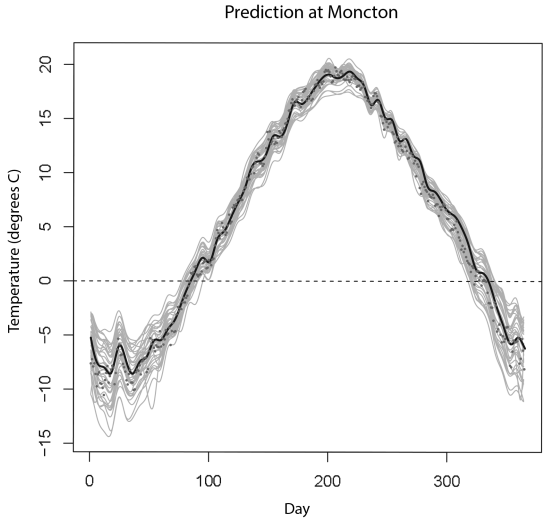

FIGURE 5: Smoothed curves by using a B-splines basis with 65 functions (*gray*), real data at Moncton weather station (red dots) and prediction at Moncton by ordinary kriging for function-value spatial data.

3.2. Using a Fourier basis

Now we use the package geofd for carrying out spatial prediction of temperature curves at ten randomly selected locations in the Canadian Maritimes Provinces (Figure 2, right panel). We use a Fourier basis with 65 functions for smoothing the data set (the same number of basis functions K as in Section 3.1) . In this example we show how the function `okfd` allows both smoothing the data and estimating directly a trace-variogram model. Posteriorly the estimation is used for performing spatial predictions of temperature curves on the ten locations already mentioned. The R code is the following

```
R> argvals<-seq(1,n, by=1)
R> col1<-sample((min(maritimes.coords[,1])*100):
(max(maritimes.coords[,1]) +  *100),10, replace=TRUE)/100
R> col2<-sample((min(maritimes.coords[,2])*100):
(max(maritimes.coords[,2]) +  *100),10, replace=TRUE)/100
R> new.coords <- cbind(col1,col2)
```

The variable `argvals` contains argument values associated with the values to be smoothed by using a Fourier basis. The variables `col1`, `col2`, and `new.coords` are used for defining the prediction locations (Figure 2, right panel). The variable `argvals` and `new.coords` are used as arguments of the function `okfd` in the code below

```
R> okfd.res<-okfd(new.coords=new.coords, coords=maritimes.coords,
+  data=maritimes.data, smooth.type="fourier", nbasis=65,
+  argvals=argvals, kappa=0.7)
```

In this example the arguments `smooth.type="fourier"` and `nbasis=65` in the function `okfd` allows us to smooth the data by using a Fourier basis with 65 functions (the number of basis functions was determined by cross-validation). In the example in Section 3.1 we use directly `cov.model="exponential"` in the function `okfd` because we chose this model previously by using the functions `trace.variog` and `fit.tracevariog`. If we do not specify a covariance model the function `okfd` estimates several models and selects the model with the least sum of squared errors. The parameter `kappa=.7` indicates that in addition to the spherical, exponential and Gaussian model, a Matern model with $\kappa = .7$ is also fitted.

A list with the objects stored in the variable `okfd.res` is obtained with the command line

```
R> names(okfd.res)

 [1] "coords"                      "data"
 [3] "argvals"                     "nbasis"
 [5] "lambda"                      "new.coords"
 [7] "emp.trace.vari"              "trace.vari"
 [9] "new.Eu.d"                    "functional.kriging.weights"
```

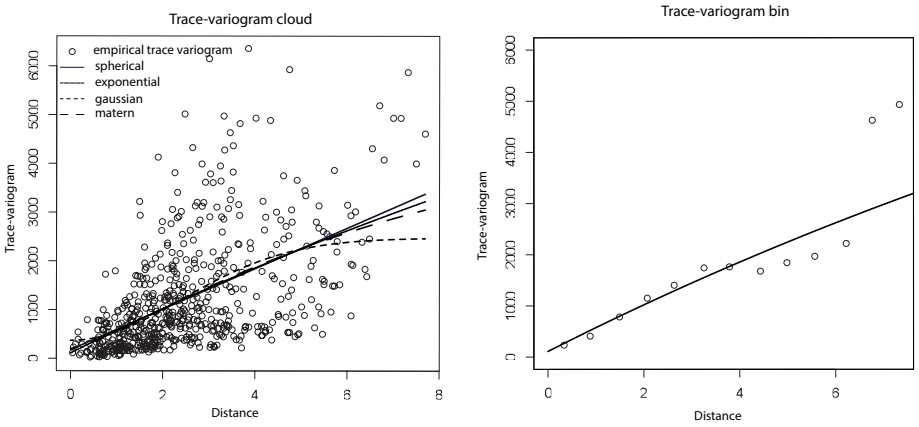

FIGURE 6: Estimated trace-variogram "cloud" and four fitted models (left panel). Estimated trace-variogram "bin" and the best fitted model (right panel).

```
[11] "krig.new.data"              "pred.var"
[13] "trace.vari.array"           "datafd"
```

We can use these objects for plotting the trace-variogram function, the estimated models and the predictions. A plot with the four fitted models and the best model is shown in Figure 6. We obtain this figure by using the command lines

```
R> plot(okfd.res, ylim=c(0,6000))
R> trace.variog.bin<-trace.variog(okfd.res$coords,
+  okfd.res$emp.trace.vari$L2norm, bin=TRUE)
R> plot(trace.variog.bin, ylim=c(0,6000), xlab="Distance",
+  ylab="Trace-variogram", main="Trace-variogram Bin")
R> lines(okfd.res$trace.vari, col=4, lwd=2)
```

Numerical results of the trace-variogram fitted models are obtained by using the command line

```
okfd.res$trace.vari.array

[[1]]
variofit: model parameters estimated by OLS (ordinary least squares):
covariance model is: spherical
parameter estimates:
      tausq     sigmasq         phi
   178.4011 644834.9056   2328.6674
Practical Range with cor=0.05 for asymptotic range: 2328.667
variofit: minimised sum of squares = 539799716
[[2]]
variofit: model parameters estimated by OLS (ordinary least squares):
```

```
covariance model is: exponential
parameter estimates:
     tausq    sigmasq        phi
  109.9118 11006.6152    23.1467
Practical Range with cor=0.05 for asymptotic range: 69.34139
variofit: minimised sum of squares = 539566326
[[3]]
variofit: model parameters estimated by OLS (ordinary least squares):
covariance model is: gaussian
parameter estimates:
    tausq   sigmasq       phi
 369.1311 2103.8708    3.3617
Practical Range with cor=0.05 for asymptotic range: 5.818404
variofit: minimised sum of squares = 552739397
[[4]]
variofit: model parameters estimated by OLS (ordinary least squares):
covariance model is: matern with fixed kappa = 0.7
parameter estimates:
    tausq   sigmasq       phi
 200.4886 4486.5365    5.8946
Practical Range with cor=0.05 for asymptotic range: 20.31806
variofit: minimised sum of squares = 541310787
```

The model with least sum of squared errors is again a exponential model (Figure 6, right panel). Consequently the function `okfd` above uses this model for solving the system in Equation 5 and for carrying out the predictions. Numerical values of predictions and prediction variances can be checked by using the commands

```
R> okfd.res[11]
R> okfd.res[12]
```

The predictions can be plotted by using the following command line

```
R>.geofd.viewer(okfd.res, argnames=c("Prediction","Day",
"Temperature"))
```

The function `.geofd.viewer` implements a `Tcl/Tk` interface (Grosjean 2010) for showing OKFD prediction results. This viewer presents two frames, the left one presents the spatial distribution of the prediction sites. The right one presents the selected prediction curve based on the point clicked by the user on the left frame. In Figure 7 we show the result of using this function. In the left panel a scatterplot with the coordinates of the prediction locations are shown. The dark point in the left panel is the clicked point and, the curve in the right panel shows the prediction at this site.

On the other hand if we want to plot all the predicted curves and analyze them simultaneously we can use the following command line

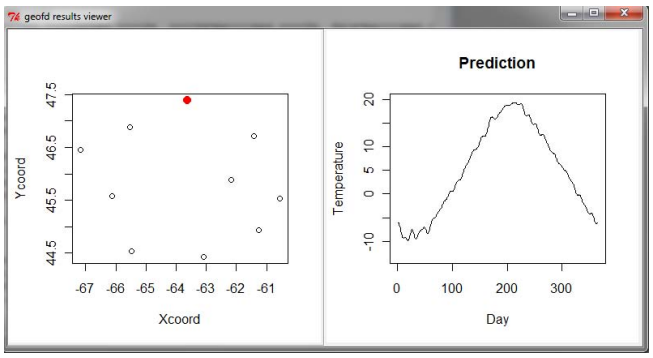

FIGURE 7: An example of the function `.geofd.viewer`. Left panel: Scatterplot with the coordinates of prediction locations. Right panel: Prediction on a clicked point (red point in left panel).

```
R> matplot(okfd.res$argvals, okfd.res$krig.new.data, col=1, lwd=1,
 type="l", +  lty=1, main="Predictions", xlab="Day",
 ylab="Temperature (degrees C)")
```

We can observe that the predicted curves (Figure 8) are consistent with the behavior of the original data set (Figure 1). This result indicates empirically that the OKFD method shows a good performance.

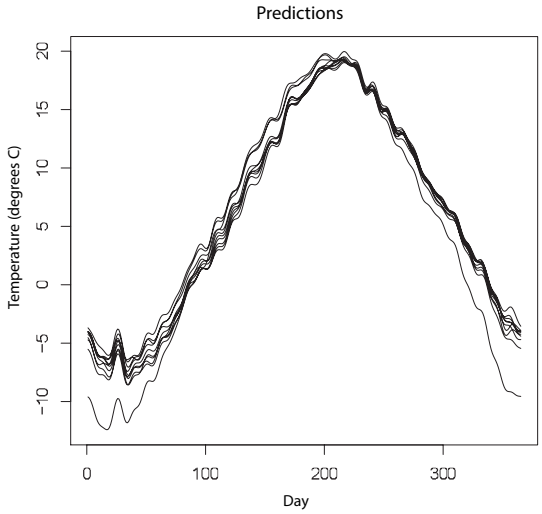

FIGURE 8: OKFD Predictions at ten randomly selected sites from Canadian Maritimes Provinces. Observed data were previously smoothed by using a Fourier basis.

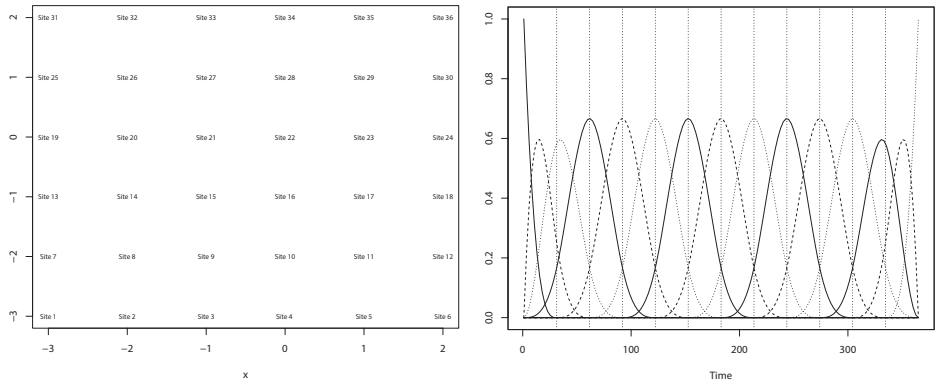

FIGURE 9: Left panel: Grid of simulated locations. Right panel: B-splines basis used in the simulation algorithm.

3.3. Using Simulated Data

In this section we discuss algorithms proposed in our package and evaluate the performance of the methodologies proposed in Section 2 by means of a simulation study.

We fixed the thirty six sites shown in Figure 9, and simulated a discretized set of spatially correlated functional data according to the model

$$X_{s_i}(t) = \sum_{l=1}^{15} a_{il} B_l(t) + \epsilon_i(t),\ i = 1, \ldots, 36 \tag{9}$$

with $\mathbf{B}(t) = (B_1(t), \ldots, B_{15}(t))$ a B-splines basis (see right panel Figure 9), a_{il}, a realization of a Gaussian random field $\mathbf{a_l} \sim N_{36}(10, \Sigma)$, where Σ is a 36×36 covariance matrix defined according to the exponential model $C(h) = 2\exp(\frac{-h}{8})$ with $h = \|s_i - s_j\|, i, j = 1, \ldots, 36$, and $\epsilon(t) \sim N_{36}(0.09, 1)$ is a random error for each fixed t, with $t = 1, \ldots, 365$. The number of basis functions and the parameters for simulating coefficients and errors were chosen empirically.

The R code for obtaining the simulated curves is the following

```
R> coordinates<-expand.grid(x= c(-3,-2, -1, 0, 1, 2),
+  y=c(-3,-2,-1,0, 1, 2))
R> mean.coef=rep(10,36)
R> covariance.coef <- cov.spatial(distance, cov.model=model,
+  cov.pars=c(2,8))
R> normal.coef=mvrnorm(15,mean.coef,covariance.coef)
R> mean.error<-rep(0, 36)
R> covariance.error <-cov.spatial(distance, cov.model=model,
+  cov.pars=c(0.09,0))
R> normal.error<-mvrnorm(365,mean.error,covariance.error)
R> argvals=seq(1, 365, len = 365)
```

```
R> nbasis=15
R> lambda=0
R> rangeval <- range(argvals)
R> norder   <- 4
R> bspl.basis <- create.bspline.basis(rangeval, nbasis,
+  norder)
R> data.basis=eval.basis(argvals, bspl.basis, Lfdobj=0)
R> func.data=t(normal.coef)%*%t(data.basis)
R> simulated.data= func.data+ normal.error
```

A plot with the simulated data and smoothed curves (by using a B-splines basis) is obtained with the following code

```
R> datafdPar <- fdPar(bspl.basis, Lfdobj=2, lambda)
R> smooth.datafd <- smooth.basis(argvals, simulated.data,
+  datafdPar)
R> simulated.smoothed=eval.fd(argvals, smooth.datafd$fd,
+  fdobj=0)
R> matplot(simulated.data, type="l", lty=1, xlab="Time",
+  ylab="Simulated data")
R> matplot(simulated.smoothed, lty=1, xlab="Time",
+  ylab="Smoothed data", type="l")
```

The simulated data are shown in the left panel of Figure 10. These data were smoothed by using a B-splines basis with 15 functions (right panel Figure 10). Once obtaining the smoothed curves we carry out a cross-validation prediction procedure. Each data location in Figure 9 is removed from the dataset and a smoothed curve is predicted at this location using OKFD based on the remaining smoothed functions.

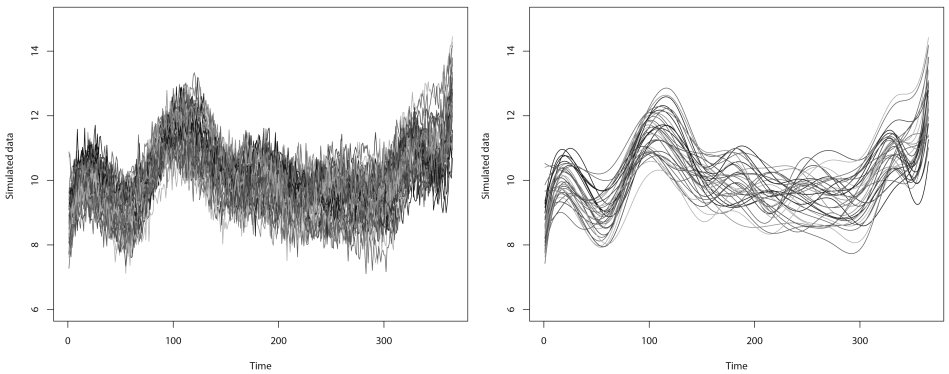

FIGURE 10: Left panel: Simulated data. Right panel: Smoothed curves (by using a B-splines basis).

The R code for obtaining the cross-validation predictions is

```
R> predictions= matrix(0, nrow=365, ncol=36)
```

```
R> for (i in 1:36)
R>     {
R>      coord.cero=matrix(coordinates[i,], nrow=1,ncol=2)
R>      okfd.res<-okfd(new.coords=coord.cero,
+       coords=coordinates[-i,], cov.model="exponential",
+       data=simulated.data[,-i], smooth.type="bsplines",
+       nbasis=15, argvals=argvals, fix.nugget=TRUE)
R>      predictions[,i]=okfd.res$krig.new.data
R>      }
```

We can plot the cross-validation predictions and the cross-validation residuals by using the following code

```
R> matplot(predictions, lty=1, xlab="Time",ylab="Predictions",
+  main="Cross-validation predictions", type="l")
R> cross.residuals=simulated.smoothed-predictions
R> matplot(cross.residuals, lty=1, xlab="Time",
+  ylab="Residuals", main="Cross-validation residuals",
+  type="l")
```

The cross-validation predictions (left panel Figure 11) shows that the predictions have the same temporal behavior as the smoothed curves (right panel Figure 10). Note also that the prediction curves have less variance. This is not surprising, because kriging is itself a smoothing method.

Figure 11 (right panel) shows cross-validation residuals. The predictions are plausible in all sites because all the residual curves are varying around zero.

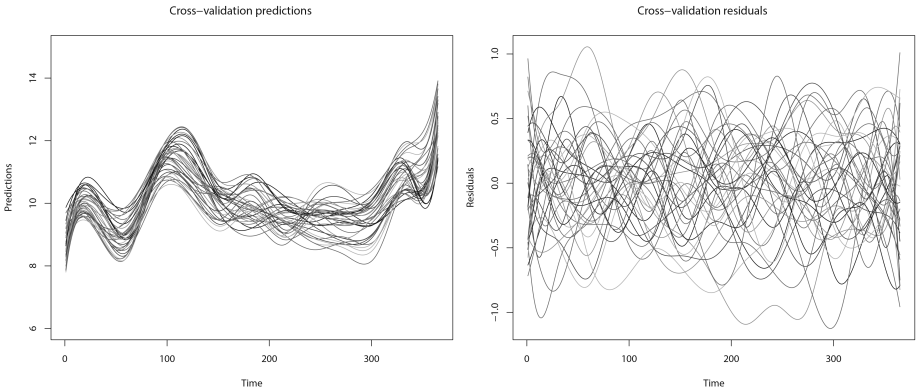

FIGURE 11: Left panel: Simulated data. Right panel: Smoothed curves (by using a B-splines basis).

The cross-validation results based on simulated data show a good performance of the proposed predictor, and indicate from a descriptive point of view that it can be adopted as a valid method for modeling spatially correlated functional data.

4. Conclusion

This paper introduces the R package geofd through an example. This package contains functions for modeling the trace-variogram function and for carrying out spatial prediction using the method of ordinary kriging for functional data. The advancements in this package would not be possible without several other important contributions to CRAN; these are reflected as geofd's package dependencies. The fda package by (Ramsay et al. 2010) provides methods for smoothing data by using basis functions. The geoR package (Ribeiro & Diggle 2001) provides functions to enable modeling the trace-variograma function. There remains scope for further extensions to geofd. We can consider other approaches for smoothing the data. For example, the use of wavelets could be useful for smoothing data with rapid changes in behavior. We plan to continue adding methods to the package. Continuous time varying kriging (Giraldo et al. 2010) and methods based on multivariable geostatistics (Giraldo 2009, Nerini et al. 2010) can be implemented in the package. However the use of these approaches could be restrictive when the number of basis functions used for smoothing the data set is large. Computationally efficient strategies are needed in this sense.

Acknowledgements

We would like to thank Andrés Pérez for his valuable contribution to upload the package geofd to CRAN. This work was partially supported by the Spanish Ministry of Education and Science through grants MTM2010-14961 and MTM2009-13985-C02-01.

[Recibido: octubre de 2011 — Aceptado: agosto de 2012]

References

Baladandayuthapani, V., Mallick, B., Hong, M., Lupton, J., Turner, N. & Caroll, R. (2008), 'Bayesian hierarchical spatially correlated functional data analysis with application to colon carcinoginesis', *Biometrics* **64**, 64–73.

Box, G. & Jenkins, G. (1976), *Time Series Analysis.*, Holden Day, New York.

Cressie, N. (1993), *Statistics for Spatial Data*, John Wiley & Sons, New York.

Cuevas, A., Febrero, M. & Fraiman, R. (2004), 'An ANOVA test for functional data.', *Computational Statistics and Data Analysis* **47**, 111–122.

Delicado, P., Giraldo, R., Comas, C. & Mateu, J. (2010), 'Statistics for spatial functional data: Some recent contributions', *Environmetrics* **21**, 224–239.

Ferraty, F. & Vieu, P. (2006), *Nonparametric Functional Data Analysis. Theory and Practice*, Springer, New York.

Giraldo, R. (2009), Geostatistical Analysis of Functional Data, PhD thesis, Universitat Politècnica de Catalunya.

Giraldo, R., Delicado, P. & Mateu, J. (2010), 'Continuous time-varying kriging for spatial prediction of functional data: An environmental application', *Journal of Agricultural, Biological, and Environmental Statistics* **15**(1), 66–82.

Giraldo, R., Delicado, P. & Mateu, J. (2011), 'Ordinary kriging for function-valued spatial data', *Environmental and Ecological Statistics* **18**(3), 411–426.

Goulard, M. & Voltz, M. (1993), Geostatistical interpolation of curves: A case study in soil science, *in* A. Soares, ed., 'Geostatistics Tróia 92', Vol. 2, Kluwer Academc Press, pp. 805–816.

Grosjean, P. (2010), *SciViews-R: A GUI API for R*, UMONS, Mons, Belgium. *http://www.sciviews.org/SciViews-R

Malfait, N. & Ramsay, J. (2003), 'The historical functional linear model', *The Canadian Journal of Statistics* **31**(2), 115–128.

MATLAB (2010), *version 7.10.0 (R2010a)*, The MathWorks Inc., Natick, Massachusetts.

Myers, D. (1982), 'Matrix formulation of co-kriging', *Mathematical Geology* **14**(3), 249–257.

Nerini, D., Monestiez, P. & Manté, C. (2010), 'Cokriging for spatial functional data', *Journal of Multivariate Analysis* **101**(2), 409–418.

R Development Core Team (2011), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
*http://www.R-project.org.

Ramsay, J., Hooker, G. & Graves, S. (2009), *Functional Data Analysis with R and MATLAB*, Springer, New York.

Ramsay, J. & Silverman, B. (2005), *Functional Data Analysis. Second edition*, Springer, New York.

Ramsay, J., Wickham, H., Graves, S. & Hooker, G. (2010), *fda: Functional Data Analysis. R package version 2.2.6.*
*http://cran.r-project.org/web/packages/fda

Ribeiro, P. & Diggle, P. (2001), 'geoR: A package for geostatistical analysis', *R-NEWS* **1**(2), 15–18.
*http://cran.R-project.org/doc/Rnews

Staicu, A., Crainiceanu, C. & Carroll, R. (2010), 'Fast methods for spatially correlated multilevel functional data', *Biostatistics* **11**(2), 177–194.

Ver Hoef, J. & Cressie, N. (1993), 'Multivariable spatial prediction', *Mathematical Geology* **25**(2), 219–240.

Wackernagel, H. (1995), *Multivariate Geostatistics: An Introduction with Applications*, Springer-Verlag, Berlin.

Yamanishi, Y. & Tanaka, Y. (2003), 'Geographically weighted functional multiple regression analysis: A numerical investigation', *Journal of Japanese Society of Computational Statistics* **15**, 307–317.