Package 'SpatFD'

June 21, 2024

Type Package

Title Functional Geostatistics: Univariate and Multivariate Functional Spatial Prediction

Version 0.0.1

Date 2024-06-18

Depends R (>= 3.6.0)

Description Performance of functional kriging, cokriging, optimal sampling and simulation for spatial prediction of functional data. The framework of spatial prediction, optimal sampling and simulation are extended from scalar to functional data. 'SpatFD' is based on the Karhunen-Loève expansion that allows to represent the observed functions in terms of its empirical functional principal components. Based on this approach, the functional auto-covariances and cross-covariances required for spatial functional predictions and optimal sampling, are completely determined by the sum of the spatial auto-covariances and cross-covariances of the respective score components. The package provides new classes of data and functions for modeling spatial dependence structure among curves. The spatial prediction of curves at unsampled locations can be carried out using two types of predictors, and both of them report, the respective variances of the prediction error. In addition, there is a function for the determination of spatial locations sampling configuration that ensures minimum variance of spatial functional prediction. There are also two functions for plotting predicted curves at each location and mapping the surface at each time point, respectively. References Bohorquez, M., Giraldo, R., and Mateu, J. (2016) <doi:10.1007/s10260-015-0340-9>, Bohorquez, M., Giraldo, R., and Mateu, J. (2016) <doi:10.1007/s00477-016-1266-y>, Bohorquez M., Giraldo R. and Mateu J. (2021) <doi:10.1002/9781119387916>.

License GPL-3

Encoding UTF-8

Imports
ggplot2,plotly,dplyr,utils,stats,proxy,reshape,gstat,sp,fda,sf,MASS,methods,geoR,tidyr,fda.usc

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Martha Patricia Bohorquez Castañeda [aut, cre],
Diego Alejandro Sandoval Skinner [aut],

Angie Villamil [aut],
Samuel Hernando Sanchez Gutierrez [aut],
Nathaly Vergel Serrano [ctb],
Miguel Angel Munoz Layton [ctb],
Valeria Bejarano Salcedo [ctb],
Venus Celeste Puertas [ctb],
Ruben Dario Guevara Gonzalez [aut],
Joan Nicolas Castro Cortes [ctb],
Ramon Giraldo Henao [aut],
Jorge Mateu [aut]

Maintainer Martha Patricia Bohorquez Castañeda <mpbohorquezc@unal.edu.co>

Repository CRAN

Date/Publication 2024-06-21 10:10:02 UTC

Contents

`AirQualityBogota` *Air quality data of Bogota, Colombia*

Description

Particular material 10 (PM10) of Bogota measure in 10 stations in the city.

Usage

```
data(AirQualityBogota)
```

Format

two data.frame of measurements 'PM10' and coordinates 'coord'. Also a map file of class SpatialPolygonDataFrame load from a shape file by `rgdal::readOGR`

Source

Secretaria de Ambiente de Bogotá

References

Monitor network of air quality of Bogota `http://rmcab.ambientebogota.gov.co`

`classification` *Classification Function for Functional Data*

Description

This function classifies new functional data based on PCA results from training data.

Usage

```
classification(data.train.pca, new.basis, k, distance, mcov = NULL)
```

Arguments

`data.train.pca`	A list of PCA results from training data.
`new.basis`	Basis object from the test data
`k`	Number of nearest neighbors to consider for classification.
`distance`	Type of distance to use (e.g., "euclidean", "mahalanobis").
`mcov`	Optional covariance matrices for Mahalanobis distance.

Value

The predicted class for the new data.

Examples

```
data(vowels)
#### Create parameters and names for the data.
p = 228 ; nelec = 21 ; nvow = 5
names_vowels = c("a","e","i","o","u")
n.basis<-c(14,13,12,13,11)
s4.gfdata = gfdata(data=vowels,p=p,names=names_vowels,coords=vowels_coords,nbasis=n.basis)
# Create train and test data
s4.sep=gfd_clasif_data(s4.gfdata, 0.8,seed = 2910)
s4.train=s4.sep$train
s4.test=s4.sep$test

# Classification
cla<-classification(data.train.pca = s4.train,
                    new.basis=s4.test[[1]]$data_fd[[1]],
                    k=4,
                    distance='euclidean',
                    mcov = mcov
)
```

COKMexico *Air quality data of Mexico*

Description

Particular material 10 (PM10) and NO2 measured in 13 and 18 locations in Mexico. The data correspond to consecutive hours from January 01, 2015 at 1:00 a.m. to May 30, 2015 at 12:00 a.m., at 23 environmental stations. The stations in the air quality network RAMA (Red Automática de Monitoreo Atmosférico), monitor hourly particulate matter up to 10 micrometers in size (PM10) and Nitrogen dioxide (NO2) among others. The particulate matter (PM) is an important component of air pollution. NO2 is a gaseous air pollutant produced by the road traffic and other fossil fuel combustion processes and it contributes to the formation and modification of other air pollutants such as particulate matter.

Usage

```
data(COKMexico)
```

Format

four data.frame of measurements 'Mex_PM10','NO2' and their coordinates 'coord_PM10' and 'coord_NO2'. Also a map (map_mex) file of class SpatialPolygonDataFrame

Source

Mexico

COKS_scores_lambdas *Functional cokriging*

Description

Linear Spatial functional prediction. Two predictors are possible: scores or lambda.

Usage

```
COKS_scores_lambdas(SFD, newcoords, model, method = "scores", fill.all=TRUE)
```

Arguments

SFD	object of class 'SpatFD'.
newcoords	The N × 2 matrix or data.frame with the spatial coordinates corresponding to the N prediction locations.
model	The linear model of coregionalization of all functional variable scores. A variogram model. A variogramModel object. See gstat package.
method	Prediction method: "scores"
fill.all	gstat function parameter. If there are more than 1 score vector and not all models or a valid and complete linear model of coregionalization is given, fill all of the direct and cross variogram model with the only model given.

Details

Each functional variable is represented in terms of its functional principal components $\chi_{\boldsymbol{s}_i}(t) = \boldsymbol{\xi}^T(t)\boldsymbol{f}_{\boldsymbol{s}_i},\ i = 1, ..., n$ where $\boldsymbol{f}_{\boldsymbol{s}_i} = \left(f^1_{\boldsymbol{s}_i}, ..., f^K_{\boldsymbol{s}_i}\right)^T$

The goal is the prediction of a spatial functional variable of $\chi^r_{\boldsymbol{s}_0}(t)$ $1 \leq r \leq P$ at the unsampled site $\boldsymbol{s}_0$ based on P spatial functional variables. The method performs cokriging directly on the scores chosen for all functional variables involved.

$\left(\boldsymbol{f}^{11}_{\boldsymbol{s}}, ..., f^{1K_1}_{\boldsymbol{s}}, ..., f^{P1}_{\boldsymbol{s}}, ..., f^{PK_P}_{\boldsymbol{s}}\right)$

Scores predictions are used to build the cokriging functional predictor.

Value

Returns a 'COKS_pred' object with functional cokriging

Author(s)

Valeria Bejarano <vbejaranos@unal.edu.co>

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

Bohorquez M.; Giraldo R. and Mateu J. Spatial prediction and optimal sampling of functional data in Geostatistical Functional Data Analysis: Theory and Methods (2021). John Wiley Sons, Chichester, UK. ISBN: 978-1-119-38784-8. https://www.wiley.com/en-us/Geostatistical+Functional+Data+Analysis-p-9781119387848.

See Also

SpatFD,summary.COKS_pred

Examples

```
data(COKMexico)
SFD_PM10_NO2 <- SpatFD(Mex_PM10, coords = coord_PM10, basis = "Fourier",
nbasis = 21, lambda = 0.000001, nharm = 2)
SFD_PM10_NO2 <- SpatFD(NO2, coords = coord_NO2, basis = "Fourier",
nbasis = 27, lambda = 0.000001, nharm = 2,
                      add = SFD_PM10_NO2)
model1 <- gstat::vgm(647677.1,"Gau",23317.05)
model1 <- gstat::vgm(127633,"Wav",9408.63, add.to = model1)
newcoords <- data.frame(x = 509926,y = 2179149)
COKS_scores_lambdas(SFD_PM10_NO2,newcoords,model1)
```

COK_crossval_loo *Leave-One-Out Cross-Validation for Functional cokriging*

Description

This function performs leave-one-out cross-validation for functional cokriging. It systematically leaves out one location at a time from the dataset, fits the model to the remaining data, and then makes a prediction for the left-out observation. It is used to assess the predictive performance of the functional cokriging model.

Usage

```
COK_crossval_loo(object, plot_show, var, show_all)
```

Arguments

- `object` — A 'COKS_pred' object obtained with function `COKS_scores_lambdas`.
- `plot_show` — A logical value. If `TRUE`, the function will generate and display a plot of the cross-validation results. If `FALSE`, no plot will be displayed. Default is `True`.
- `var` — A numerical value indicating the number of the variable to be used in the cross-validation. Default is `1`.
- `show_all` — A logical value. If `TRUE`, the function will display all graphs at once, else it will display them one by one. Default is `FALSE`.

Value

An object containing the results of the leave-one-out cross-validation. Includes:

- `performance_metrics` — Summary statistics describing the overall predictive performance, such as mean squared error.
- `plots` — The generation of plots showing the cross-validation results, controlled by the `plot_show` parameter. If `plot_show` is `TRUE`, this will contain the plots; otherwise, it will be empty.

Author(s)

Venus Puertas <vpuertasg@unal.edu.co>

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Optimal sampling for spatial prediction of functional data. Statistical Methods & Applications, 25(1), 39-54.

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

See Also

`recons_fd`, `KS_scores_lambdas`

Examples

```
# Load the data and the required packages
library(SpatFD)
library(gstat)
data(COKMexico)

# Create the SpatFD data objects
SFD_PM10_NO2 <- SpatFD(Mex_PM10, coords = coord_PM10, basis = "Fourier",
                       nbasis = 21, lambda = 0.000001, nharm = 2,
                       name = names(Mex_PM10))
SFD_PM10_NO2 <- SpatFD(NO2, coords = coord_NO2, basis = "Fourier",
                       nbasis = 27, lambda = 0.000001, nharm = 2,
```

```
                          add = SFD_PM10_NO2,name = names(NO2))

# Fit the model
model1 <- gstat::vgm(647677.1,"Gau",23317.05)
model1 <- gstat::vgm(127633,"Wav",9408.63, add.to = model1)

# Perform the cokriging
newcoords <- data.frame(x = 509926,y = 2179149)
coks <- COKS_scores_lambdas(SFD_PM10_NO2,newcoords,model1)

# Perform the cross-validation along NO2
COK_crossval_loo(object = coks, var = 2,show_all=TRUE)
```

`coord` *Coordinates of measurement stations Bogota, Colombia*

Description

Coordinates of 10 stations in the city.

Usage

```
data(AirQualityBogota)
```

Format

data.frame of coordinates

Source

Secretaria de Ambiente de Bogotá

References

Monitor network of air quality of Bogota `http://rmcab.ambientebogota.gov.co`

`coord_NO2` *Coordinates of air quality data of Mexico*

Description

18 locations in Mexico where measured NO2.

Usage

```
data(COKMexico)
```

Format

data.frame of coordinates 'coord_NO2'

Source

Mexico

`coord_PM10` *Coordinates of air quality data of Mexico*

Description

13 locations in Mexico where measured Mex_PM10.

Usage

```
data(COKMexico)
```

Format

data.frame of coordinates 'coord_PM10'

Source

Mexico

`create_mcov` *Create Covariance Matrices given a series of spatial model parameters*

Description

This function creates covariance matrices for spatial data based on the provided model parameters.

Usage

```
create_mcov(coordenadas, t.models)
```

Arguments

`coordenadas`	A matrix of coordinates.
`t.models`	A data frame with model parameters.

Value

A list of covariance matrices.

Examples

```
  data(vowels)
  t.models = data.frame(sill = c(1887.47,1447.27,3533.96,1850.27,432.63),
                        phi  = c(106.68,109.1,19.4,150.32,31.52),
                        tausq =c(0,0,0,0,0),
                        model = as.character(c("gaussian","gneiting",
                                               "matern","cubic",
                                               "wave")),
                        kappa = c(0,0,7.3,0,0)

)
create_mcov(vowels_coords,t.models)
```

`CrossSpatFD` *Creates univariate and multivariate CrossSpatFD object to perform crossed validation for functional spatial prediction.*

Description

Creates univariate and multivariate CrossSpatFD object considering the base od a SpatFD or FD object to perform crossed validation for functional spatial prediction.

Usage

```
CrossSpatFD(data,coords,basis,lambda=0,nharm=NULL,name=NULL,add=NULL,...)
```

Arguments

`data`	Data must be provided in a data-frame or a matrix where each column corresponds to a location, and the rows are a sequence of data points, that is, the rows are ordered according to time, frequency, depth, ... Data can also be an fd-object from the fda package.
`coords`	A data-frame or a matrix with spatial coordinates (x,y). The number of columns in data must coincide with the number of rows in coords for each variable.
`basis`	The basis from the SpatFD or FD object.
`lambda`	The value of the smoothing parameter.
`nharm`	The number of harmonics or eigenfunctions to be reported in the Functional Principal Components results if vp is not given.
`name`	A new name for data can be assigned.
`add`	Other variables can be added for spatial multivariate functional prediction, that is, for functional cokriging. It is not necessary that all variables are observed at the same spatial locations.
`...`	arguments from fda create.bspline.basis or create.fourier.basis.

Details

The CrossSpatFD-objects storage the functional data, its parameters, the functional principal component analysis results, and the spatial coordinates for each variable. Each variable has its own functional data, data-frame or matrix and its spatial coordinates file.

Value

For each variable: The functional data and functional principal components linked with spatial coordinates.

Note

1. This function is for internal use and should not be implemented directly

Author(s)

Diego Sandoval <diasandovalsk@unal.edu.co> & Angie Villamil <acvillamils@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Optimal sampling for spatial prediction of functional data. Statistical Methods & Applications, 25(1), 39-54.

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

See Also

`summary.SpatFD`

`crossval_loo` *Leave-One-Out Cross-Validation for Functional Kriging*

Description

This function performs leave-one-out cross-validation for functional kriging and cokriging. It systematically leaves out one observation at a time from the dataset, fits the model to the remaining data, and then makes a prediction for the left-out observation. It is used to assess the predictive performance of the functional kriging model.

Usage

```
crossval_loo(object, plot_show)
```

Arguments

`object` A 'KS_pred' object obtained with function `KS_scores_lambdas`.

`plot_show` A logical value. If `TRUE`, the function will generate and display a plot of the cross-validation results. If `FALSE`, no plot will be displayed. Default is `True`.

Value

An object containing the results of the leave-one-out cross-validation. Includes:

`performance_metrics`
: Summary statistics describing the overall predictive performance, such as mean squared error.

`plots` The generation of plots showing the cross-validation results, controlled by the `plot_show` parameter. If `plot_show` is `TRUE`, this will contain the plots; otherwise, it will be empty.

Author(s)

Joan Castro <jocastroc@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Optimal sampling for spatial prediction of functional data. Statistical Methods & Applications, 25(1), 39-54.

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

See Also

`recons_fd`, `KS_scores_lambdas`

Examples

```
# Example code demonstrating how to use the crossval_loo function
library(SpatFD)
library(gstat)

# Load data and coordinates
data(AirQualityBogota)

#s_0 nonsampled location. It could be data.frame or matrix and one or more locations of interest
newcoorden=data.frame(X=seq(93000,105000,len=100),Y=seq(97000,112000,len=100))
#newcoorden=data.frame(X=110000,Y=126000)
#newcoorden=matrix(c(110000.23,109000,109500,130000.81,129000,131000),nrow=3,ncol=2,byrow=TRUE)

# Building the SpatFD object
SFD_PM10 <- SpatFD(PM10, coords = coord[, -1], basis = "Bsplines",
nbasis = 17,norder=5, lambda = 0.00002, nharm=3)
```

```
# Semivariogram models for each spatial random field of scores
modelos <- list(vgm(psill = 2199288.58, "Wav", range = 1484.57, nugget =  0),
                vgm(psill = 62640.74, "Mat", range = 1979.43, nugget = 0,kappa=0.68),
                vgm(psill =37098.25, "Exp", range = 6433.16, nugget =  0))

# Functional kriging. Functional spatial prediction at each location of interest
#method = "lambda"
#Computation of lambda_i
KS_SFD_PM10_l <- KS_scores_lambdas(SFD_PM10, newcoorden ,method = "lambda",
model = modelos)
# method = "scores"
#Simple kriging of scores
KS_SFD_PM10_sc <- KS_scores_lambdas(SFD_PM10, newcoorden, method = "scores", model = modelos)
# method = "both"
KS_SFD_PM10_both <- KS_scores_lambdas(SFD_PM10, newcoorden, method = "both", model = modelos)

# Cross Validation
crossval_loo(KS_SFD_PM10_l)
crossval_loo(KS_SFD_PM10_sc)
crossval_loo(KS_SFD_PM10_both)
```

`FD_optimal_design` *Optimal Spatial Design For Functional Data*

Description

Given a variogram model and a set of points in which we want to predict certain variable optimally, this function finds where must be placed the stations in which the information will be collected for functional or scalar data.

Usage

```
FD_optimal_design(k, s0, model, fixed_stations = NULL,
                    scalar = FALSE, nharm = NULL,
                    method = "lambda", grid = NULL,
                    map = NULL, plt = FALSE)
```

Arguments

`k`	The number of new stations that are going to be located.
`s0`	`matrix`, `array`, `data.frame`, or `SpatialPoints` object that contains the coordinates of the points in which we want to optimally predict using functional kriging. Each row corresponds to each point. If you want to predict optimally in a certain area, you must create a random sample large enough of points of that area and pass it to this function.
`model`	A `VariogramModel` from `gstat` package or a list of `VariogramModels` objects if a distinct model is going to be used for each harmonic. For scalar data only one model is needed.

fixed_stations	If there are already some stations on the field that are not going to be removed, here should be passed their coordinates. The object must be of class `matrix`, `array`, `data.frame`, `SpatialPoints`, `SpatFD`, or `NULL` if there are no fixed stations.
scalar	Boolean that indicates if the optimization is for functional data (FALSE) or scalar data (TRUE). If TRUE, nharm is set to 1.
nharm	Number of harmonics of the functional principal components that will be used in the prediction. If it is not specified it will be set to the number of models passed, then this parameter shouldn't be specified for scalar data.
method	Functional kriging method that will be used. Currently available "lambda" and "scores". See details bellow.
grid	Coordinates in which the new stations can be located. `grid` must be of type `matrix`, `array`, `data.frame`, `SpatialPoints`. If you don't pass `grid`, you must pass `map`; if you pass both, `map` will be used.
map	Spatial object from sp package such as `Line`, `Lines`, `Polygon`, `SpatialPolygons`, `SpatialGrid` or `SpatialPixels` in which the new stations will be located. This object will also be used for creating the plot.
plt	`TRUE` or `FALSE`. If `TRUE`, a nice `ggplot2` plot with the result will be generated. Se example bellow.

Details

Bohorquez, M., Giraldo, R., Mateu, J. (2016) present several methods for finding the best combination predictor-design according to the kriging prediction error variance for functional data. They show different functional kriging methods and two of them are implemented on this function.

If method is "lambda", optimal spatial sampling using FPCA and simple kriging will be used (see section 3.2 of Bohorquez, M., Giraldo, R., Mateu, J. (2016)). If method is "scores", simple kriging will be applied on each harmonic and the total variance will be minimized. This total variance is computed as follows:

$$TotVar = \sum_{j=1}^{nharm} V_j$$

where V_j is the variance of the simple kriging prediction of j-th score.

Value

The function returns an `OptimalSpatialDesign` object that is a list with the following elements:

new_stations	`matrix` `array` object with the coordinates of the new stations.
fixed_stations	`matrix` `array` object with the coordinates of the fixed stations.
plot	`ggplot2` plot.

Warning

When method is 'lambda', the minimized value is not the variance, but the negative of expression (12) in Bohorquez, M., Giraldo, R., & Mateu, J. (2016), that is

$$-\sum_{l=1}^{L} \varsigma_l' \Omega^{-1} \varsigma_l$$

Note

'lambda' method tends to be faster than 'scores' method.

Author(s)

Nathaly Vergel Serrano <nvergel@unal.edu.co> & Samuel Sánchez Gutiérrez <ssanchezgu@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Optimal sampling for spatial prediction of functional data. Statistical Methods & Applications, 25(1), 39-54.

See Also

print.OptimalSpatialDesign

Examples

```
library(gstat)
s0 <- cbind(2*runif(100),runif(100)) # random coordinates on (0,2)x(0,1)
fixed_stations <- cbind(2*runif(4),runif(4))
x_grid <- seq(0,2,length = 30)
y_grid <- seq(0,1,length = 30)
grid <- cbind(rep(x_grid,each = 30),rep(y_grid,30))
model  <- vgm(psill = 5.665312,
                  model = "Exc",
                  range = 8000,
                  kappa = 1.62,
                  add.to = vgm(psill = 0.893,
                               model = "Nug",
                               range = 0,
                               kappa = 0))
FD_optimal_design(k = 10, s0 = s0, model = model,
                  grid = grid, nharm = 2, plt = TRUE,
                  fixed_stations = fixed_stations) -> OSD
OSD$new_stations
OSD$fixed_stations
OSD$plot
class(OSD)
```

`generate_basis` *Creates functional ortogonal basis as fd object.*

Description

This function returns the first nth elements of a functional basis as an fd object.

Usage

```
generate_basis(basis = "Fourier",n_functions = 10,L = NULL,fda_basis = NULL)
```

Arguments

`basis`	Name of the functional basis. Currently only `"Fourier"` or `"Legendre"` are supported.
`n_functions`	Positive integer giving the number of functions that are going to be generated.
`L`	For `"Fourier"` basis, limits of the domain of the basis $(-L, L)$. For `"Legendre"` basis, the domain is $(-1, 1)$.
`fda_basis`	`basisfd` object from package `fda` that will be used in the `fd` object that is returned. If `NULL`, default basis are generated.

Details

Fourier basis functions are given by:

$$f_k(x) = \sqrt{\frac{2}{L}}\sin\left(\frac{k\pi x}{2L}\right)$$

for $k = 2, 4, 6, ...$, and

$$f_k(x) = \sqrt{\frac{2}{L}}\cos\left(\frac{(k+1)\pi x}{2L}\right)$$

for $k = 1, 3, 5, ...$

Furthermore, Legendre basis functions are given by:

$$f_k(x) = \frac{1}{2^n n!}\frac{d^n}{dx}(x^2 - 1)^n$$

for $k = 1, 2, 3, 4, ...$

Value

`fda::fd` object with `n_functions` curves.

Note

Generating n Legendre basis functions requires to evaluate $\frac{n(n+1)}{2}$ derivates, so its recomended to use values below 10.

Author(s)

Samuel Sánchez Gutiérrez <ssanchezgu@unal.edu.co>.

References

Conway, J. B. (2019). A course in functional analysis (Vol. 96). Springer.

See Also

sim_functional_process

Examples

```
library(fda)

# 10 Fourier functions
res <- generate_basis(L=1)
plot(res)

# 20 Fourier functions
res <- generate_basis(n_functions = 20,L = 3)
plot(res)

# 10 Legendre functions
res <- generate_basis(basis = "Legendre")
plot(res)

# 7 Legendre functions
res <- generate_basis(basis = "Legendre", n_functions = 7)
plot(res)
```

---

gfdata *Creates gfdata objects.*

Description

Creates an object of the class gfdata from spatial coordinates, and functions or time-series observed at each spatial location. Time series is a generic term. In fact, observations might be across the frequency or across another spatial dimension such as depth, instead of time.

Usage

```
gfdata(data, p, basis = "Bsplines", coords = NULL, nbasis = NULL,
            names = NULL, lambda = 0)
```

Arguments

data	Data must be provided in a matrix where each column corresponds to a subject, and the rows are a sequence of data points, that is, the rows are ordered according to time, frequency, depth, ... Also must include a column with classes for classification in the last column
p	Number of repetitions for each class
basis	Basis functions. "Fourier" or "Bsplines". By default, "Bsplines".
coords	A matrix with spatial coordinates (x,y).
nbasis	The number of basis functions.
names	Names for the data classes.
lambda	The value of the smoothing parameter.

Details

The gfdata-objects storage the functional data, its parameters, the functional principal component analysis results, and the spatial coordinates for each variable. Each variable has its own functional data, data-frame or matrix and its spatial coordinates file.

Value

For each subject and class: The functional data and functional principal components linked with spatial coordinates.

Author(s)

Venus Puertas <vpuertasg@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Optimal sampling for spatial prediction of functional data. Statistical Methods & Applications, 25(1), 39-54.

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

See Also

summary.gfdata

Examples

```
library(SpatFD)
data(vowels)

#### Create parameters and names for the data.
p = 228 ; nelec = 21 ; nvow = 5
names_vowels = c("a","e","i","o","u")
n.basis<-c(14,13,12,13,11)
```

```
s4.gfdata = gfdata(data=vowels,p=p,names=names_vowels,coords=vowels_coords,nbasis=n.basis)
```

gfd_clasif_data *Divide the data in train and test dataset*

Description

This function divides the data in train and test datasets

Usage

```
gfd_clasif_data(gfd_data, prop.train, seed = NULL)
```

Arguments

gfd_data	Object of class 'gfdata'. Not NULL.
prop.train	Number between 0 and 1, indicating the proportion to be left on train dataset
seed	seed for the sampling algorithms

Value

gfdata divided object

Author(s)

Diego Sandoval <diasandovalsk@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

Examples

```
library(SpatFD)
data(vowels)

#### Create parameters and names for the data.
p = 228 ; nelec = 21 ; nvow = 5
names_vowels = c("a","e","i","o","u")
n.basis<-c(14,13,12,13,11)

s4.gfdata = gfdata(data=vowels,p=p,names=names_vowels,coords=vowels_coords,nbasis=n.basis)
s4.sep=gfd_clasif_data(s4.gfdata, 0.8,seed = 2910)
```

```
s4.train=s4.sep$train
s4.test=s4.sep$test
```

---

gfd_variog_geoR *Generate Variograms for Functional Data from a gfdata object*

Description

This function generates variograms for functional data based on PCA results.

Usage

```
gfd_variog_geoR(gfd_pca_Data, pairsmin = 2)
```

Arguments

gfd_pca_Data	A list of PCA results for functional data.
pairsmin	Minimum number of pairs for variogram calculation.

Value

A list containing geodata objects and variograms.

Examples

```
data(vowels)

#### Create parameters and names for the data.
p = 228 ; nelec = 21 ; nvow = 5
names_vowels = c("a","e","i","o","u")
n.basis<-c(14,13,12,13,11)
s4.gfdata = gfdata(data=vowels,p=p,names=names_vowels,coords=vowels_coords,nbasis=n.basis)

s4.sep=gfd_clasif_data(s4.gfdata, 0.8,seed = 2910)

s4.train=s4.sep$train

s4.var.geoR=gfd_variog_geoR(s4.train)
```

ggmap_KS *Map plot of a 'KS_pred' object*

Description

A visualization of the predicted kriging in a colormap for a specific window time.

Usage

```
ggmap_KS(KS, map_path, window_time = NULL, method = "lambda", map_n = 5000,
zmin = NULL, zmax = NULL, graph = "plotly")
```

Arguments

`KS`	Object of class 'KS_pred'. Not NULL.
`map_path`	Character indicating the directory of the shape file or an object of class SpatialPolygonDataFrame load from a shape file by `rgdal::readOGR`. If NULL the prediction is plot in a rectangle. Default NULL.
`window_time`	numeric. Vector of window time to see the spatial prediction. If NULL choose the range values of KS$SFD. Default NULL.
`method`	character. "lambda" or "scores". Default "lambda".
`map_n`	numeric. Number of points to sample in the map. Default 5000.
`zmin`	numeric. Minimum value predicted for the color scale. If NULL is chosen from the data. Default NULL.
`zmax`	numeric. Maximum value predicted for the color scale. If NULL is chosen from the data. Default NULL.
`graph`	character. "plotly" or "gg" whether to use plotly or ggplot graphics. Default "plotly".

Value

Plotly or ggplot image

Author(s)

Diego Sandoval <diasandovalsk@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

See Also

KS_scores_lambdas

Examples

```
library(gstat)
data(AirQualityBogota)
newcoorden=data.frame(X=seq(93000,105000,len=100),Y=seq(97000,112000,len=100))

SFD_PM10 <- SpatFD(PM10, coords = coord[, -1], basis = "Bsplines", nbasis = 17,
norder=5, lambda = 0.00002, nharm=3)

modelos <- list(vgm(psill = 2634000, "Exp", range = 2103.25, nugget =  0),
                vgm(psill = 101494.96, "Exp", range = 1484.57, nugget = 0),
                vgm(psill =53673, "Exp", range = 42406, nugget =  0))

KS_SFD_PM10_both <- KS_scores_lambdas(SFD_PM10, newcoorden, method = "both",
model = modelos)

ggmap_KS(KS_SFD_PM10_both,
         map_path = map,
         window_time = c(5108,5109,5110),
         method = "scores",
         zmin = 50,
         zmax = 120)
```

`ggplot_KS` *ggplot of predicted functions*

Description

Plot with or without predicted variance in each spatial location of the functional kriging.

Usage

```
ggplot_KS(KS, show.varpred = FALSE, main = "Functional Data",
main2 = "Functional Data", ylab = "Value", xlab = "Time", ndigits = 2,
palette.plot = c("#440154FF", "#3336FF", "#33FCFF", "#33FF4C", "#FDE725FF"))
```

Arguments

`KS`	Object of class 'KS_pred'. Not NULL.
`show.varpred`	Boolean. If the predicted variance is shown. Default FALSE.
`main`	character. Title of the plot.Default "Functional Data".
`main2`	character. If there are two methods where used, the title of the second plot. Default "Functional Data".
`ylab`	character. Name of the y-axis.
`xlab`	character. Name of the x-axis.
`ndigits`	numeric. Number of decimals for the predicted variance if shown. Default 2.
`palette.plot`	list. String values of hexadecimal colors. Default c("#440154FF", "#3336FF", "#33FCFF", "#33FF4C", "#FDE725FF")

Value

ggplot. If there are two plots a list of ggplots.

Author(s)

Diego Sandoval <diasandovalsk@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

See Also

KS_scores_lambdas

Examples

```
library(gstat)
data(AirQualityBogota)
newcoorden=data.frame(X=seq(93000,105000,len=10),Y=seq(97000,112000,len=10))
SFD_PM10 <- SpatFD(PM10, coords = coord[, -1], basis = "Bsplines", nbasis = 17,
norder=5, lambda = 0.00002, nharm=3)
modelos <- list(vgm(psill = 2634000, "Exp", range = 2103.25, nugget =  0),
                vgm(psill = 101494.96, "Exp", range = 1484.57, nugget = 0),
                vgm(psill =53673, "Exp", range = 42406, nugget =  0))

KS_SFD_PM10_both <- KS_scores_lambdas(SFD_PM10, newcoorden, method = "both",
model = modelos)

ggplot_KS(KS_SFD_PM10_both,show.varpred = FALSE,
          main = "Plot 1 - Using Scores",
          main2 = "Plot 2 - Using Lambda",
          ylab = "PM10")
```

KS_scores_lambdas *Functional Kriging*

Description

Linear Spatial functional prediction. Two predictors are possible: scores or lambda.

Usage

```
KS_scores_lambdas(SFD, newcoords, model, method = "lambda", name = NULL, fill.all = NULL)
```

Arguments

`SFD`	object of class 'SpatFD'.
`newcoords`	The N × 2 matrix or data.frame with the spatial coordinates corresponding to the N prediction locations.
`model`	The scores variogram model. A variogramModel object. See gstat package.
`method`	Prediction method: "lambda" or "scores". By default "lambda". See details.
`name`	The variable to predict in SpatFD object. By default, the predictions is performed for the first variable in the SpatFD object.
`fill.all`	gstat function parameter. If there are more than 1 score vector and not all models or a valid and complete linear model of coregionalization is given, fill all of the direct and cross variogram model with the only model given.

Details

"lambda" option corresponds to the predictor $\breve{\chi}_{\boldsymbol{s}_0}(t)$, given by

$$\breve{\chi}_{\boldsymbol{s}_0}(t) = \sum_{i=1}^{n} \lambda_i \chi_{\boldsymbol{s}_i}(t)$$

and weigths are found such that minimize $\|\boldsymbol{\chi}_{\boldsymbol{s}_0}(t) - \breve{\chi}_{\boldsymbol{s}_0}(t)\|^2$.

"scores" method performs kriging or cokriging directly on the scores and predictions are used to build the functional prediction

It is used simple cokriging to predict the vector $\boldsymbol{f}(\boldsymbol{s}_0) = (f_1(\boldsymbol{s}_0), ..., f_K(\boldsymbol{s}_0))^T$ at the unsampled location $\boldsymbol{s}_0$. The predictor is $f^*(\boldsymbol{s}_0)$, so the prediction of the curve $\chi_{\boldsymbol{s}_0}(t)$ is $\chi^*_{\boldsymbol{s}_0}(t) = \boldsymbol{\xi}^T(t)\boldsymbol{f}^*(\boldsymbol{s}_0), \quad i = 1, ..., n$.

Value

Returns a 'KS_pred' object with functional kriging: weights (lambda) using the first method and kriging score predictions using the second method in Bohorquez, M., Giraldo, R., & Mateu, J. (2016).

Author(s)

Diego Sandoval <`diasandovalsk@unal.edu.co`> & Angie Villamil <`acvillamils@unal.edu.co`>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Optimal sampling for spatial prediction of functional data. Statistical Methods & Applications, 25(1), 39-54.

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

Bohorquez M.; Giraldo R. and Mateu J. Spatial prediction and optimal sampling of functional data in Geostatistical Functional Data Analysis: Theory and Methods (2021). John Wiley Sons, Chichester, UK. ISBN: 978-1-119-38784-8. https://www.wiley.com/en-us/Geostatistical+Functional+Data+Analysis-p-9781119387848.

See Also

SpatFD,summary.KS_pred

Examples

```
library(gstat)
data(AirQualityBogota)

newcoorden=data.frame(X=110000,Y=125000)

# Recibir los datos, suavizarlos y ACP
SFD_PM10 <- SpatFD(PM10, coords = coord[,2:3], basis = "Bsplines",
nbasis = 17,norder=5, lambda = 0.00002, nharm=3)

#Variogram model for each component
modelos <- list(vgm(psill = 2634000, "Exp", range = 2103.25, nugget =  0),
                vgm(psill = 101494.96, "Exp", range = 1484.57, nugget = 0),
                vgm(psill =53673, "Exp", range = 42406, nugget =  0))

#Genera los scores y los lambdas para predecir en nuevas coordenadas

#method = "lambda"
KS_SFD_PM10_l <- KS_scores_lambdas(SFD_PM10, newcoorden ,method = "lambda",
model = modelos)
class(KS_SFD_PM10_l)

#method = "scores"
KS_SFD_PM10_sc <- KS_scores_lambdas(SFD_PM10, newcoorden, method = "scores",
model = modelos)

#method = "both"
KS_SFD_PM10_both <- KS_scores_lambdas(SFD_PM10, newcoorden, method = "both",
model = modelos)
```

---

map *map of Bogota, Colombia*

Description

Map of Bogota.

Usage

```
data(AirQualityBogota)
```

Format

Map file of class SpatialPolygonDataFrame

Source

Unidad Administrativa Especial de Catastro Distrital

References

https://datosabiertos.bogota.gov.co/dataset/sector-catastral

---

map_mex *map of Mexico*

Description

Map of Mexico.

Usage

```
data(COKMexico)
```

Format

Map file of class SpatialPolygonDataFrame

Source

Automatic Monitoring System SEDEMA

References

http://www.aire.df.gob.mx/default.php

---

mclass_data *Get the mean of means for each class*

Description

This function generates multivariate vowel data based on the provided mean functions and basis functions.

Usage

```
mclass_data(mean.mean, n.basis, type.basis = "bspline")
```

Arguments

mean.mean	A list of means for each class.
n.basis	A list of basis functions for each class.
type.basis	Type of basis functions to use (e.g., "bspline", "fourier").

Value

A data frame containing the generated vowel data.

Examples

```
  data(vowels)

#### Create parameters and names for the data.
p = 228 ; nelec = 21 ; nvow = 5
names_vowels = c("a","e","i","o","u")
n.basis<-c(14,13,12,13,11)
s4.gfdata = gfdata(data=vowels,p=p,names=names_vowels,coords=vowels_coords,nbasis=n.basis)
s4.sep=gfd_clasif_data(s4.gfdata, 0.8,seed = 2910)
s4.train=s4.sep$train
s4.test=s4.sep$test
mean_mean <- mean_mean(s4.train)
class_mean <- mclass_data(mean_mean,n.basis)
```

---

mean_mean — *Calculate Mean Functions for Each Class*

Description

This function calculates the mean functions for each class based on PCA results from training data.

Usage

```
mean_mean(data.train.pca)
```

Arguments

`data.train.pca` A list of PCA results from training data.

Value

A list of mean functions for each class.

Examples

```
  data(vowels)
#### Create parameters and names for the data.
p = 228 ; nelec = 21 ; nvow = 5
names_vowels = c("a","e","i","o","u")
n.basis<-c(14,13,12,13,11)
s4.gfdata = gfdata(data=vowels,p=p,names=names_vowels,coords=vowels_coords,nbasis=n.basis)

s4.sep=gfd_clasif_data(s4.gfdata, 0.8,seed = 2910)
```

```
s4.train=s4.sep$train
mean_mean <- mean_mean(s4.train)
```

Mex_PM10 *Air quality data of Mexico*

Description

Particular material 10 (PM10) measured in 13 locations in Mexico.

Usage

```
data(COKMexico)
```

Format

data.frame of measurements 'Mex_PM10'

Source

Mexico

NO2 *Air quality data of Mexico*

Description

NO2 measured in 18 locations in Mexico.

Usage

```
data(COKMexico)
```

Format

data.frame of measurement 'NO2'

Source

Mexico

PM10 — *PM10 of Bogota, Colombia*

Description

Particular material 10 (PM10) of Bogota measure in 10 stations in the city.

Usage

```
data(AirQualityBogota)
```

Format

data.frame of measurements 'PM10'

Source

Secretaria de Ambiente de Bogotá

References

Monitor network of air quality of Bogota `http://rmcab.ambientebogota.gov.co`

print.OptimalSpatialDesign — *Print of OptimalSpatialDesign objects*

Description

This functions prints a summary of the main objects of OptimalSpatialDesign objects.

Usage

```
## S3 method for class 'OptimalSpatialDesign'
print(x, ...)
```

Arguments

`x`	Object of class 'OptimalSpatialDesign'.
`...`	arguments from print

Value

Shows the amount of fixed stations, new stations and the first six new coordinates.

Author(s)

Samuel Sánchez Gutiérrez <ssanchezgu@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Optimal sampling for spatial prediction of functional data. Statistical Methods & Applications, 25(1), 39-54.

See Also

FD_optimal_design

Examples

```
library(gstat)
data(AirQualityBogota)

vgm_model  <- gstat::vgm(psill = 5.665312,
                  model = "Exc",
                  range = 8000,
                  kappa = 1.62,
                  add.to = vgm(psill = 0.893,
                               model = "Nug",
                               range = 0,
                               kappa = 0))

my.CRS <- sp::CRS("EPSG:21899") # https://epsg.io/21899
map <- as(map, "Spatial")

bogota_shp <- sp::spTransform(map,my.CRS)
target <- sp::spsample(bogota_shp,n = 100, type = "random")
# The set of points in which we want to predict optimally.
old_stations <- sp::spsample(bogota_shp,n = 3, type = "random")
# The set of stations that are already fixed.

FD_optimal_design(k = 10, s0 = target,model = vgm_model,
               map = map,plt = TRUE,#method = "scores",
               fixed_stations = old_stations) -> res
print(res)
```

---

recons_fd *Linear combinations for functional kriging*

Description

This is an internal function for functional kriging and cokriging. To perform the linear combinations to obtain functional kriging and cokriging. Once optimization process is finished and scores or lambda are obtained, this function builds the prediction, performing the linear combination between coefficients and basis functions.

Usage

```
recons_fd(X,name = NULL)
```

Arguments

X	A 'KS_pred' or 'COKS_pred' object obtained with function KS_scores_lambdas or COKS_scores_lambdas
name	Name of the variable of interest. Default 1, the first variable.

Value

Spatial functional predictions. The fd object based on the two functional kriging methods described in Bohorquez, M., Giraldo, R., & Mateu, J. (2016). List of

fd_scores

fd_lambda

Author(s)

Diego Sandoval <diasandovalsk@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Optimal sampling for spatial prediction of functional data. Statistical Methods & Applications, 25(1), 39-54.

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

Bohorquez M.; Giraldo R. and Mateu J. Spatial prediction and optimal sampling of functional data in Geostatistical Functional Data Analysis: Theory and Methods (2021). John Wiley Sons, Chichester, UK. ISBN: 978-1-119-38784-8. https://www.wiley.com/en-us/Geostatistical+Functional+Data+Analysis-p-9781119387848. ~

See Also

KS_scores_lambdas

Examples

```
library(gstat)
data(AirQualityBogota)
newcoorden=data.frame(X=110000,Y=125000)
SFD_PM10 <- SpatFD(PM10, coords = coord[,2:3], basis = "Bsplines", nbasis = 17,
norder=5, lambda = 0.00002, nharm=3)
modelos <- list(vgm(psill = 2634000, "Exp", range = 2103.25, nugget =  0),
                vgm(psill = 101494.96, "Exp", range = 1484.57, nugget = 0),
                vgm(psill =53673, "Exp", range = 42406, nugget =  0))

#method = "lambda"
```

```
KS_SFD_PM10_l <- KS_scores_lambdas(SFD_PM10, newcoorden ,method = "lambda",
model = modelos)
curves_PM10_l <- recons_fd(KS_SFD_PM10_l)
plot(curves_PM10_l)

#method = "scores"
KS_SFD_PM10_sc <- KS_scores_lambdas(SFD_PM10, newcoorden, method = "scores",
model = modelos)
curves_PM10_sc <- recons_fd(KS_SFD_PM10_sc)
plot(curves_PM10_sc)
```

`scores` *Spatial random field of scores*

Description

Creates a list of P data-frames, one for each of the P spatial functional variables in the SpatFD-object. Each data-frame contains one column for each harmonic selected with its respective n_p-score values, and two additional columns for horizontal and vertical spatial coordinates.

Usage

```
scores(X)
```

Arguments

`X`	An object of class SpatFD.

Details

For a SpatFD object with P spatial functional variables measured on n_p locations each, and nharm=k_p, p=1,..., P. The scores-function builds a data-frame with n_p rows and k_p+2 columns. The first two columns are the horizontal and vertical spatial coordinates x,y. The rest of the columns are the score values for each of the first k_p harmonics selected for the p variable, at each of the n_p locations.

Value

A list with P data-frames, with dimension $n_p \times k_p + 2$, each. p=1,...,P.

Note

Functional principal components (FPCA) are applied to the centered data. The scores are scalar second order stationary Random fields, in virtue of the requirements for FPCA. Hence, the covariance function exists, that is, models are bounded, and always have sill and range parameters. Of course, some models have additional parameters, such as smoothing parameters. From the theoretical perspective, in this case, there is no possibility of non-bounded variogram models. The spatial covariance between two curves is determined by the sum of the covariance between spatial score vectors associated, see Bohorquez, Giraldo and Mateu 2016 and Bohorquez, Giraldo and Mateu

2021. This covariance can be modeled using the usual packages for geostatistical analysis, such as geoR and gstat. Finally, the sill of the variogram model for each dimension is bounded for the respective eigenvalue. Actually, an adequate option is to use this eigenvalue as sill and estimate the rest of the parameters.

Author(s)

Diego Sandoval <diasandovalsk@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Optimal sampling for spatial prediction of functional data. Statistical Methods & Applications, 25(1), 39-54.

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

See Also

SpatFD

Examples

```
data(AirQualityBogota)
newcoorden=data.frame(X=seq(93000,105000,len=100),Y=seq(97000,112000,len=100))
# Building the SpatFD object
SFD_PM10 <- SpatFD(PM10, coords = coord[, -1], basis = "Bsplines", nbasis = 17,
norder=5, lambda = 0.00002, nharm=3)
scores(SFD_PM10)
```

---

sim_functional_process

Simulation of unconditional or conditional functional spatial process.

Description

Given a variogram model, this functions simulates several realizations of a functional spatial process. This simulation can be conditioned to observed data.

Usage

```
sim_functional_process(nsims,variograms,nbasis,coords,data = NULL,
                       data_coords = NULL,basis = NULL,mu = NULL,L = NULL)
```

Arguments

nsims	Integer giving the number of curves to simulate.
variograms	gstat::gstatVariogram or list of them giving the variogram model for each score. If only one is provided, it will be recycled.
nbasis	Integer giving the number of basis functions on which the process is going to be projected.
coords	Gridded sp::SpatialPoints or sp::SpatialPixels, or array coordinates of the curves that are going to be simulated.
data	fda::fd object containing the observed curves for conditional simulation. If data is not provided, inconditional simulation is performed.
data_coords	sp::SpatialPoints or array coordinates of the observed data.
basis	Character giving the basis of functions (only for inconditional simulation) (nbasis must be provided).
mu	fda::fd object of the mean function of the process, default is zero. Only used in unconditional simulation.
L	Limits of the symetric interval centered on zero that is the domain of the basis that is going to be created in unconditional simulation case.

Details

When data is passed, conditional simulation is performed. That means that each simulated realization of the process interpolated the observed curves in data. If data is NULL, the realizations of the process are simulated without imterpolation restrictions.

Value

A list of nsims SpatFD objects each one with as much curves as points are in coords.

Author(s)

Samuel Sánchez Gutiérrez <ssanchezgu@unal.edu.co>.

References

Bohorquez, M., Giraldo, R. & Mateu, J. Multivariate functional random fields: prediction and optimal sampling. Stoch Environ Res Risk Assess 31, 53–70 (2017). https://doi.org/10.1007/s00477-016-1266-y

See Also

generate_basis

Examples

```
library(gstat)
library(fda)
library(sp)

data("CanadianWeather")
canada.CRS <- CRS("+init=epsg:4608")
coords <- SpatialPoints(CanadianWeather$coordinates,
                        proj4string = CRS("+init=epsg:4326"))
coords <- spTransform(coords,canada.CRS)
obs <- CanadianWeather$dailyAv[,,1] # Temperature

Lfd_obj <- int2Lfd(m = 2)
create.bspline.basis(rangeval = c(1,365),
                     nbasis = 40, norder = 4) -> mi.base
mi.fdPar <- fdPar(mi.base, Lfd_obj, lambda = 7.389)
mi.fd <- smooth.basis(argvals = 1:365,
                      y = obs, fdParobj = mi.fdPar)

nbasis <- 5
canada <- mi.fd$fd
canada.pca <- pca.fd(canada,nharm = 10)
base_ort <- canada.pca$harmonics[1:nbasis]
canada_mean <- canada.pca$meanfd

formula2fd <- function(rango, expresion) {
  # Generate grid
  n <- 500  # length of the grid
  x <- seq(rango[1], rango[2], length.out = n)

  # evaluate expression on the grid
  y_vals <- eval(parse(text = expresion))

  # convert to fd
  basis <- create.bspline.basis(rangeval = rango, nbasis = 30)
  fd_obj <- Data2fd(x, y_vals,basisobj = basis)

  return(fd_obj)
}
media <- formula2fd(c(-1,1),"3*sin(x*4)")

# No conditional
vario <- vgm(.25, "Exp", .5, .05)
nbasis <- 6
sims <- sim_functional_process(10,vario,nbasis,coords,basis = 'Legendre',mu = media)
class(sims)
length(sims)

class(sims[[1]])
# plot(sims[[3]][[1]]$data_fd)

sims <- sim_functional_process(10,vario,nbasis,coords,basis = 'Legendre')
```

```
class(sims)
length(sims)
class(sims[[1]])
# plot(sims[[3]][[1]]$data_fd)

# Conditional
vario <- vgm(100, "Exp", 900, 10)
new_coords <- spsample(coords,100,type = "regular")
gridded(new_coords) <- TRUE
length(new_coords)
a <- sim_functional_process(10,vario,nbasis,new_coords,canada,coords)
class(a)
length(a)
class(a[[1]])
#plot(a[[1]][[1]]$data_fd)

vario <- vgm(100, "Wav", 900, 10)
a <- sim_functional_process(10,vario,nbasis,new_coords,canada,coords)
class(a)
length(a)
class(a[[1]])
#plot(a[[1]][[1]]$data_fd)
```

SpatFD *Creates univariate and multivariate SpatFD objects.*

Description

Creates an object of the class SpatFD from spatial coordinates, and functions or time-series observed at each spatial location. Time series is a generic term. In fact, observations might be across the frequency or across another spatial dimension such as depth, instead of time.

Usage

```
SpatFD(data, coords, basis = "Bsplines", nbasis = 4, lambda = 0, nharm = NULL,
name = NULL, add = NULL, ...)
```

Arguments

data	Data must be provided in a data-frame or a matrix where each column corresponds to a location, and the rows are a sequence of data points, that is, the rows are ordered according to time, frequency, depth, ... Data can also be an fd-object from the fda package.
coords	A data-frame or a matrix with spatial coordinates (x,y). The number of columns in data must coincide with the number of rows in coords for each variable.
basis	Basis functions. "Fourier" or "Bsplines". By default, "Bsplines".
nbasis	The number of basis functions.

`lambda`	The value of the smoothing parameter.
`nharm`	The number of harmonics or eigenfunctions to be reported in the Functional Principal Components results if vp is not given.
`name`	A new name for data can be assigned.
`add`	Other variables can be added for spatial multivariate functional prediction, that is, for functional cokriging. It is not necessary that all variables are observed at the same spatial locations.
`...`	arguments from fda create.bspline.basis or create.fourier.basis.

Details

The SpatFD-objects storage the functional data, its parameters, the functional principal component analysis results, and the spatial coordinates for each variable. Each variable has its own functional data, data-frame or matrix and its spatial coordinates file.

Value

For each variable: The functional data and functional principal components linked with spatial coordinates.

Note

1. Although there is no limit for the number of variables for functional cokriging, the real limitation is found on the constraints required to find a valid multivariate covariance model. So, it is highly recommended to apply the parsimony principle.

2. Locations must be in the same region of interest to make sense to include all of them in the same prediction model. However, each variable can be observed in different spatial locations and each can have a different number of observations. There is no limit for the number of variables to be included in this object.

Author(s)

Diego Sandoval <`diasandovalsk@unal.edu.co`> & Angie Villamil <`acvillamils@unal.edu.co`>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Optimal sampling for spatial prediction of functional data. Statistical Methods & Applications, 25(1), 39-54.

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

See Also

`summary.SpatFD`

Examples

```
# Load data
data(AirQualityBogota)

# Create an univariate object using 2 nharm
SFD_PM10 <- SpatFD(PM10, coords = coord[,2:3], basis = "Bsplines", nbasis = 91,
lambda = 0.00002, nharm = 2)
SFD_PM10
```

---

summary.COKS_pred *Summary of COKS_pred objects*

Description

This functions shows a summary of the main objects of COKS_pred objects.

Usage

```
  ## S3 method for class 'COKS_pred'
summary(object, ...)
```

Arguments

`object`	Object of class 'COKS_pred'.
`...`	arguments from summary

Value

This functions prints according to method computed: eigenvalues, variance of prediction and each of the models.

Author(s)

Joan Nicolás Castro <jocastroc@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

See Also

COKS_scores_lambdas

Examples

```
data(COKMexico)
SFD_PM10_NO2 <- SpatFD(Mex_PM10, coords = coord_PM10,
basis = "Fourier", nbasis = 21, lambda = 0.000001, nharm = 2)
SFD_PM10_NO2 <- SpatFD(NO2, coords = coord_NO2,
basis = "Fourier", nbasis = 27, lambda = 0.000001,
nharm = 2,add = SFD_PM10_NO2)
model1 <- gstat::vgm(647677.1,"Gau",23317.05)
model1 <- gstat::vgm(127633,"Wav",9408.63, add.to = model1)
newcoords <- data.frame(x = 509926,y = 2179149)
coks <- COKS_scores_lambdas(SFD_PM10_NO2,newcoords,model1)
summary(coks)
```

`summary.gfdata` *Summary of gfdata objects*

Description

This functions shows a summary of the main objects of gfdata objects.

Usage

```
## S3 method for class 'gfdata'
summary(object, ...)
```

Arguments

`object`	Object of class 'gfdata'.
`...`	arguments from summary.

Value

For each variable included in the gfdata object, this functions return: Head of data, Coordinates, Eigenvalues, Mean coefficients, Proportion of explained variance by each component

Author(s)

Joan Nicolás Castro <jocastroc@unal.edu.co>, Venus Celeste Puertas <vpuertasg@unal.edu.co>

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

See Also

gfdata

Examples

```
data(vowels)

#### Create parameters and names for the data.
p = 228 ; nelec = 21 ; nvow = 5
names_vowels = c("a","e","i","o","u")
n.basis<-c(14,13,12,13,11)
s4.gfdata = gfdata(data=vowels,p=p,names=names_vowels,coords=vowels_coords,nbasis=n.basis)
summary.gfdata(object=s4.gfdata)
```

---

summary.KS_pred *Summary of KS_pred objects*

Description

This functions shows a summary of the main objects of KS_pred objects.

Usage

```
  ## S3 method for class 'KS_pred'
summary(object, ...)
```

Arguments

object	Object of class 'KS_pred'.
...	arguments from summary

Value

This functions prints according to method computed: eigenvalues, variance of prediction and each of the models.

Author(s)

Joan Nicolás Castro <jocastroc@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

See Also

KS_scores_lambdas

Examples

```
library(gstat)
data(AirQualityBogota)

newcoorden=data.frame(X=110000,Y=125000)

# Recibir los datos, suavizarlos y ACP
SFD_PM10 <- SpatFD(PM10, coords = coord[,2:3], basis = "Bsplines", nbasis = 17,
norder=5, lambda = 0.00002, nharm=3)

#Variogram model for each component
modelos <- list(vgm(psill = 2634000, "Exp", range = 2103.25, nugget =  0),
                vgm(psill = 101494.96, "Exp", range = 1484.57, nugget = 0),
                vgm(psill =53673, "Exp", range = 42406, nugget =  0))

#Genera los scores y los lambdas para predecir en nuevas coordenadas

#method = "lambda"
KS_SFD_PM10_l <- KS_scores_lambdas(SFD_PM10, newcoorden ,method = "lambda",
model = modelos)
summary(KS_SFD_PM10_l)
```

---

summary.SpatFD *Summary of SpatFD objects*

Description

This functions shows a summary of the main objects of SpatFD objects.

Usage

```
  ## S3 method for class 'SpatFD'
summary(object, ...)
```

Arguments

object	Object of class 'SpatFD'.
...	arguments from summary.

Value

For each variable included in the SpatFd object, this functions return: Head of data, Coordinates, Eigenvalues, Mean coefficients, Proportion of explained variance by each component

Author(s)

Joan Nicolás Castro <jocastroc@unal.edu.co>.

References

Bohorquez, M., Giraldo, R., & Mateu, J. (2016). Multivariate functional random fields: prediction and optimal sampling. Stochastic Environmental Research and Risk Assessment, 31, pages53–70 (2017).

See Also

SpatFD

Examples

```
# Load data
data(AirQualityBogota)

# Create an univariate object using 2 nharm
SFD_PM10 <- SpatFD(PM10, coords = coord[,2:3], basis = "Bsplines", nbasis = 91,
lambda = 0.00002, nharm = 2)
summary(SFD_PM10)
```

---

vowels — *Imaginary thinking of the five Spanish vowels*

Description

Data consist of EEG signals taken from 21 electrodes from imaginary thinking of the five Spanish vowels, to be applied into a BCI for a hand prosthesis.

Usage

```
data(vowels)
```

Format

two data.frame of measurements 'vowels' and coordinates 'vowels_coords'.

Source

https://github.com/carlos-sarmientov/DATABASE-IMAGINED-VOWELS-1

References

Classification techniques for imaginary speech brain signal through spatial functional data `https://repositorio.unal.edu`

`vowels_coords` *Coordinates of electrodes from the vowels data set*

Description

Coordinates of the electrodes from the vowels data set.

Usage

```
data(vowels)
```

Format

matrix of coordinates

Source

https://github.com/carlos-sarmientov/DATABASE-IMAGINED-VOWELS-1

References

Classification techniques for imaginary speech brain signal through spatial functional data `https://repositorio.unal.edu`

Index